

(19) 世界知的所有権機関  
国際事務局(43) 国際公開日  
2004 年 6 月 10 日 (10.06.2004)

PCT

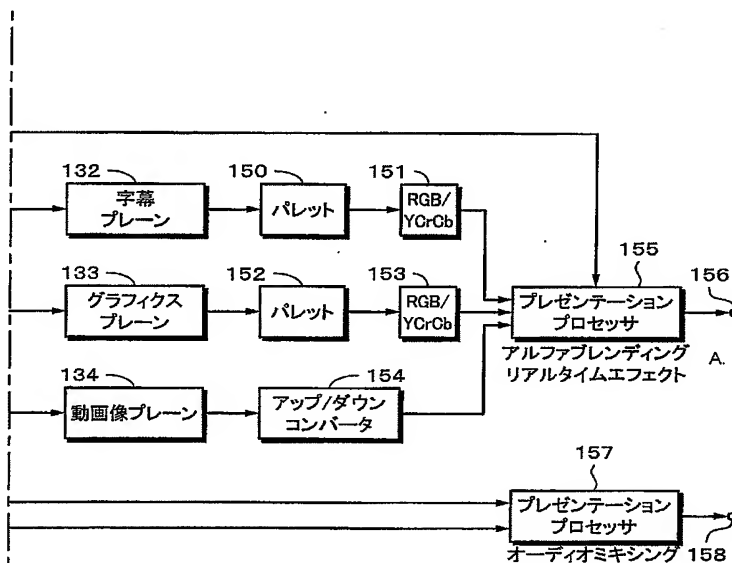
(10) 国際公開番号  
WO 2004/049710 A1

- (51) 国際特許分類<sup>7</sup>: H04N 5/93 特願2003-22551 2003 年 1 月 30 日 (30.01.2003) JP  
特願2003-74441 2003 年 3 月 18 日 (18.03.2003) JP
- (21) 国際出願番号: PCT/JP2003/014511 (71) 出願人 (米国を除く全ての指定国について): ソニー株式会社 (SONY CORPORATION) [JP/JP]; 〒141-0001 東京都品川区北品川 6 丁目 7 番 3 5 号 Tokyo (JP).
- (22) 国際出願日: 2003 年 11 月 14 日 (14.11.2003)
- (25) 国際出願の言語: 日本語 (72) 発明者; および
- (26) 国際公開の言語: 日本語 (75) 発明者/出願人 (米国についてのみ): 浜田 俊也 (HAMADA, Toshiya) [JP/JP]; 〒141-0001 東京都品川区北品川 6 丁目 7 番 3 5 号 ソニー株式会社内 Tokyo (JP). 加藤 元樹 (KATO, Motoki) [JP/JP]; 〒141-0001 東京都品川区北品川 6 丁目 7 番 3 5 号 ソニー株式会社内 Tokyo (JP).
- (30) 優先権データ:  
特願 2002-346133  
2002 年 11 月 28 日 (28.11.2002) JP

[続葉有]

(54) Title: REPRODUCTION DEVICE, REPRODUCTION METHOD, REPRODUCTION PROGRAM, AND RECORDING MEDIUM

(54) 発明の名称: 再生装置、再生方法、再生プログラムおよび記録媒体



132...CAPTION PLANE  
150...PALETTE  
133...GRAPHICS PLANE  
152...PALETTE  
134...MOVING PICTURE PLANE  
154...UP/DOWN CONVERTER  
155...PRESENTATION PROCESSOR  
A...ALPHA BLENDING REAL TIME EFFECT  
157...PRESENTATION PROCESSOR  
158...AUDIO MIXING

(57) Abstract: A user interface having a high degree of freedom is realized in a large-capacity disc-shaped recording medium containing a content. Button image data, audio data on corresponding sound effect, and control instructions of audio data and video data are multiplexed on the stream and recorded on a disc. Scenario reproduced from the disc is stored in a buffer (104) and video data and audio data are respectively accumulated in a buffer (105). The video data is read out from the buffer (105) according to the scenario and spread on a graphics plane (133). Together with this, corresponding audio data is read out of the buffer (105) and reproduced by a sound player (106D). The sound effect is reproduced in accordance with modification of the button image based on the button state change, thereby realizing more effective user interface.

(57) 要約: コンテンツ記録済み大容量ディスク状記録媒体において、自由度の高いユーザインターフェイスを実現する。ボタン画像データおよび対応する効果音の音声データと、音声データや画像データの制御命令とをストリームに多重化してディスクに記録する。ディスクから再生したシナリオをバッファ 104 に格納し、画像データおよび音声データをバッファ 105 に夫々溜め込む。シナリオに従いバッファ 105 から画像データを読み出され

してグラフィクスプレーン 133 に展開する。それと共に、対応する音声データがバッファ 105 から読み出されサウンドプレーヤ 106D により再生

[続葉有]

WO 2004/049710 A1



(74) 代理人: 杉浦 正知, 外(SUGIURA, Masatomo et al.);  
〒171-0022 東京都豊島区南池袋2丁目49番7号 池袋  
パークビル7階 Tokyo (JP).

添付公開書類:  
— 国際調査報告書

2文字コード及び他の略語については、定期発行される  
各PCTガゼットの巻頭に掲載されている「コードと略語  
のガイダンスノート」を参照。

(81) 指定国 (国内): CN, JP, US.

## 明 細 書

再生装置、再生方法、再生プログラムおよび記録媒体

## 技術分野

- 5      この発明は、ブルーレイディスク (Blu-ray Disc) といった大容量の記録媒体に記録されたプログラムに対するユーザによるインタラクティブな操作を可能とする再生装置、再生方法、再生プログラムおよび記録媒体に関する。

## 10   背景技術

- 近年、記録可能で記録再生装置から取り外し可能なディスク型記録媒体の規格として、Blu-ray Disc (ブルーレイディスク) 規格が提案されている。Blu-ray Disc 規格では、記録媒体として直径 12 cm、カバー層 0.1 mm のディスクを用い、光  
15   学系として波長 405 nm の青紫色レーザ、開口数 0.85 の対物レンズを用いて、最大で 27 GB (ギガバイト) の記録容量を実現している。これにより、日本の BS デジタルハイビジョン放送を、画質を劣化させることなく 2 時間以上記録することが可能である。

- この記録可能光ディスクに記録する AV (Audio/Video) 信号のソース (供給源) としては、従来からの、例えばアナログテレビジョン放送によるアナログ信号によるものと、例えば BS デジタル放送をはじめとするデジタルテレビジョン放送によるデジタル信号によるものとが想定されている。Blu-ray Disc 規格では、これらの放送による AV 信号を記録する方法を定めた規格は、既に作られ  
20   ている。  
25   ている。

現状の Blu-ray Disc の派生規格として、映画や音楽な

どが予め記録された、再生専用の記録媒体を開発する動きが進んでいる。映画や音楽を記録するためのディスク状記録媒体としては、既にDVD (Digital Versatile Disc) が広く普及しているが、このBlu-ray Discの規格に基づいた再生専用光ディスクは、Blu-ray Discの大容量および高速な転送速度などを活かし、ハイビジョン映像を高画質なままで2時間以上収録できる点が、既存のDVDとは大きく異なり、優位である。

現状のBlu-ray Discの規格では、ディスクに記録されている映像コンテンツの一覧を画面表示する方法や、その一覧表上にカーソルを表示させ、再生したい映像コンテンツをユーザに選択させるなどといったユーザインターフェイスに関する機能が定められていない。これらの機能は、Blu-ray Discに対する記録再生を行う記録再生装置本体によって実現されている。そのため、同一の記録媒体を再生した場合でも、再生に用いた記録再生装置によってコンテンツ一覧画面のレイアウトが異なってしまい、ユーザインターフェイスにも差が生じ、必ずしもユーザにとって使い易いものではない。再生専用ディスクとしては、再生機器によらず、ディスク（コンテンツ）制作者が意図した通りのメニュー画面などが表示され、意図通りのユーザインターフェイスが実現される必要がある。

映像コンテンツの再生中に選択画面が表示され、ユーザの選択によってストーリーが分岐していくマルチストーリーの機能は、一般にインタラクティブ機能とも呼ばれる。このインタラクティブ機能を実現するためには、ディスク制作者が再生順序や分岐を定めたシナリオを作り、そのシナリオをプログラム言語、スクリプト言語等を使って記述し、ディスクに記録しておく必要がある。再生装置側では、そのプログラムを読み込み、実行することで、制作者の意図に従った映像コ



ンテンツの再生や、分岐のための選択画面提示を実現することになる。  
。

現状のBlu-ray Disc規格(Blu-ray Disc Rewritable Format Ver1.0)では、この制作者の意図通りのユーザインターフェイ  
5 スを実現するための、メニュー画面や分岐選択画面の構成方法、ユーザ入力に対する処理を記述する方法が定められていない。現状では、Blu-ray Discを用いて、制作者が意図したシナリオ通りの再生を、再生装置の製造メーカーや機種に左右されことなく互換性を持たせた形で実現することが難しいという問題点があった。

10 再生専用ディスクにおいては、同一の被写体を複数のカメラで撮影し、ユーザが好みのアングルでの視聴を選択できるマルチアングル機能が制作者側より求められており、これを実現できる仕組みを用意する必要がある。

再生専用ディスクにおいては、字幕を表示する仕組みが不可欠である。  
15 る。しかしながら、この字幕表示についても、現状のBlu-ray Disc規格では、定められていない。

従来から、例えばDVD(Digital Versatile Disc)の規格においては、上述のようなインタラクティブな機能が既に実現されていた。DVDビデオでは、動画を再生中にリモートコントロールコマンドなど  
20 を用いてメニュー画面を呼び出し、メニュー画面上に配置されたボタンを選択するなどして、再生場面を変更するなどの処理が可能であった。字幕を表示する仕組みも規定されていた。字幕表示については、予め用意されている日本語字幕と英語字幕とを切り換えて表示させることができた。マルチアングル機能も実現されていた。

25 DVDの場合、メニュー画面を固定的なサブピクチャデータにより構成し、メニュー画面が呼び出された際に、動画データにこのサブピ

クチャデータを合成して表示する。特開平10-308924号公報に、このように動画データにサブピクチャデータを合成して記録可能なDVDに記録する構成が記載されている。

- 従来技術によるメニュー画面の一例の表示について、概略的に説明
- 5 する。例えば、映画が収録されたDVDを再生装置で再生する際に、映画本編が再生されるのに先立って、メニュー画面が表示される。一般に、メニュー画面には、複数のボタンが配置される。各ボタンには、それぞれ特定の動作が割り当てられており、ボタンを選択し実行を指示すると、選択されたボタンに割り当てられた動作が実行される。
- 10 「映画本編」と表示されたボタンを選択し実行を指示することで、当該ボタンに割り当てられた動作が実行され、当該DVDに収録された映画の本編が再生されることになる。

- ユーザは、リモートコントロールコマンド（以下、リモコンと略称する）に設けられた、上下左右の方向をそれぞれ指示するようにされた
- 15 たキー（方向キー）などを操作して、メニュー画面に表示されたボタンの一つを選択状態とし、そこで決定キーを操作することで、選択されたボタンに割り当てられた動作を実行状態とする。ボタンには、通常（非選択）状態、選択状態及び実行状態という3種類の状態が存在し、ユーザが区別し易いように、それぞれ画像や色などが異ならされ
- 20 ている。一般に、選択状態および実行状態のボタンは、同時には一つしか存在しない。

- DVDビデオの場合、ボタンの表示は、それぞれサブピクチャ、ハイライトと呼ばれる2種類のデータにより実現される。第1図は、従来技術による一例のDVDメニュー画面300を示す。メニュー画面
- 25 300は、「タイトルメニュー」と名付けられ、それぞれ「本編再生」、「映像特典」および「音声設定」と表示された3個のボタン30

1 A、3 0 1 Bおよび3 0 1 Cが配置される。この第1図の例では、  
「本編再生」ボタン3 0 1 Aの外枠の色がボタン3 0 1 Aの元の外枠  
の色と異なった色にされており、「本編再生」ボタン3 0 1 Aが選択  
状態となっていることが示される。

- 5 この状態で、ユーザがリモコンの方向キーを所定に操作することにより、第2図A、第2図Bおよび第2図Cにそれぞれ例示されるように、選択状態のボタンを変更することができる。選択されたボタンは、上述の第1図と同様に、外枠の色が非選択状態の場合に対して異ならされる。第1図の状態、ユーザがリモコンに設けられた決定ボタン  
10 を操作すると、その時点で選択状態にある「本編再生」ボタン3 0 1 Aが第3図に一例が示されるように、実行状態を表す色に変化される。その後、メニュー画面3 0 0が消え、本編が再生されることになる。以上が、DVDビデオによるボタンの基本動作である。

- ところで、第1図に例示されるメニュー画面3 0 0は、第4図A、  
15 第4図Bおよび第4図Cにそれぞれ一例が示されるように、背景画像3 1 0、サブピクチャ3 1 1およびハイライト3 1 2の3種類のデータから構成される。背景画像3 1 0は、静止画や、当該DVDに収録されるコンテンツ本体の動画などが表示される。

- サブピクチャ3 1 1は、第5図に例示されるように、1ピクセル当  
20 たり2ビットの情報で表現された1枚のビットマップ画像と、4色の色情報(A 0, B 0, C 0, D 0)と、当該サブピクチャ3 1 1の表示開始位置を示す座標(X, Y)とを有する。色情報A 0、B 0、C 0およびD 0は、それぞれ1組のR(赤)G(緑)B(青)データからなる1色の色情報データであって、R、G、B各色について各々8  
25 ビットの情報を有する。ビットマップ画像は、1ピクセルにつき2ビットの情報を有し、2ビットを用いて上述の4色の色情報(A 0, B

0, C 0, D 0)の中から1色をピクセル毎に選択する。色情報は、透明度情報も含まれており、サブピクチャ311に対して背景画像310が透けて見える領域を設けることができる。サブピクチャ311の表示位置は、サブピクチャ311の左上隅の位置が背景画像310  
5 に対する相対的な座標(X, Y)で示される。

サブピクチャ311は、表示開始および停止時刻を示す情報、当該サブピクチャ311に対してフェードイン、フェードアウトといった視覚効果を施すコマンドを有することができる。

DVDビデオにおいては、同時に複数のビットマップ画像を表示する  
10 ことができない。そのため、上述した第1図に例示されるような、複数のボタンが配置されたようなメニュー画面300の表示は、第4図Bに例示されるような、3個のボタン画像を含む1枚の大きなビットマップ画像によりなされる。第4図Bのサブピクチャ311のビットマップ画像において、ボタン301A、301Bおよび301Cの  
15 外側の領域を透明領域として設定することで、このサブピクチャ311と背景画像310とを合成した際に、ボタン301A、301Bおよび301Cの表示領域以外に背景画像310を透過させて表示させることができる。

ハイライト312は、サブピクチャ311に用いられている4色を  
20 、他の4色に置き換えるための情報である。第5図に例示されるように、ハイライト312は、色情報として、選択状態の色情報(A1, B1, C1, D1)および実行状態の色情報(A2, B2, C2, D2)を有する。これらの色情報も、上述のサブピクチャ311と同様、RGBそれぞれ8ビットで表現される4色の色情報である。

25 また、ハイライト312は、位置情報として、色を置き換える領域の座標の組を有する。色を置き換える範囲は、サブピクチャ311の

画像全体に限らず、サブピクチャ 3 1 1 内の一部の矩形領域とすることが  
できる。このハイライト 3 1 2 により色を変化させるサブピクチャ 3 1 1  
内の矩形領域の数がユーザが選択可能なボタンの数に相当する。それぞ  
れの矩形領域の表示位置は、左上隅および右下隅の位置が  
5 座標 (X, Y) で示される。例えば、ボタン 3 0 1 A に対応するハイ  
ライト 3 1 2 A の位置は、座標 (X 1, Y 1) および (X 1', Y 1')  
で示される。ボタン 3 0 1 B および 3 0 1 C にそれぞれ対応するハイ  
ライト 3 1 2 B および 3 1 2 C も、同様である。

一例として、ハイライト 3 1 2 A は、背景画像 3 1 0 の座標 (X 1  
10, Y 1) および (X 1', Y 1') で示される領域の色情報 (A 0, B  
0, C 0, D 0) を、選択状態の色として指定される色情報 (A 1,  
B 1, C 1, D 1) に置き換える。このとき、背景画像 3 1 0 の色情  
報 A 0 がハイライト 3 1 2 A の色情報 A 1 に置き換えられる。同様に  
、背景画像 3 1 0 の色情報 B 0 が色情報 B 1 に、色情報 C 0 が色情報  
15 C 1 に、色情報 D 0 が色情報 D 1 に、それぞれ置き換えられる。

このハイライト 3 1 2 による色の置き換えの例を、第 1 図、第 2 図  
A、第 2 図 B および第 2 図 C、ならびに、第 3 図を用いて上述した、  
メニュー画面 3 0 0 におけるボタン 3 0 1 A の状態に応じた表示変化  
に対応させて説明する。ボタン 3 0 1 A が非選択状態のときには、ボ  
タン 3 0 1 A の枠、表面および文字がそれぞれ色情報 B 0、色情報 C  
20 0 および色情報 D 0 を用いて表示されているものとする。ボタン 3 0  
1 A を選択状態とすると、ボタン 3 0 1 A の枠色 B 0 がハイライト 3  
1 2 A の選択状態の対応する色情報 B 1 で置き換えられる。表面色 C  
0 および文字色 D 0 は、変化しない。次いで、ボタン 3 0 1 A が実行  
25 状態とされると、選択状態の色に対して、ボタン 3 0 1 A の表面色 C  
0 が色情報 C 1 に置き換えられる。枠色 B 1 および文字色 D 0 は、選

択状態に対して変化しない。

- DVDビデオにおいて、通常の映像再生時は、背景画像310に相当する画像だけが表示されている。字幕付きの映画などの場合は、映画本編が再生されている背景画像310と、字幕が表示されているサブピクチャ311とが合成された状態で表示がなされている。

しかしながら、サブピクチャ311、選択状態を表すハイライト312、実行状態を表すハイライト312は、それぞれ最大4色しか用いることができず、色数の多いサブピクチャを表示することができないという問題点があった。

- 10 ハイライト312は、サブピクチャ311の色を置き換え変更するだけであったため、選択状態や実行状態においてボタン中に表示されている文字を変えることや、ボタンの形状が変化するような効果を実現することができず、表現力に乏しいという問題点があった。

- 字幕表示とボタン表示とを、サブピクチャ311を用いた同一の仕組みで実現していたため、字幕とボタンとをそれぞれ独立して表示制御することや、字幕あるいはボタン表示に透明度を設定してこれらを重ね合わせて表示させるといった、合成処理もできないという問題点があった。

- 20 メニュー画面を呼び出した際に、背景に表示される動画データの再生が中断されていた。このように、例えインタラクティブな機能が実現されていても、その機能を実現するためのユーザインターフェイスの自由度が低いという問題点があった。

- 字幕の表示や変化に同期させて効果音を発生させる仕組みが規格で定められていないため、字幕に同期させて効果音を発生させるようなことができないという問題点があった。

ボタンについても、ユーザがボタンを選択状態にした際に発せられ

る効果音や、ボタンの選択状態時に決定キーを操作するなどして、当該ボタンを実行状態にした際に発せられるクリック音のような効果音を発生させる仕組みが規格で定められていなかったため、表現豊かなユーザインターフェイスを実現することが困難であるという問題点が  
5 あった。

自由度の高いユーザインターフェイスを実現する上で、ボタンの描画速度や更新速度、ユーザ入力に対する応答性が問題となるが、それらを見積もるためのグラフィックスのデコーダモデルが必要であった。

ここでいう効果音とは、動画プレーン上に表示される動画や静止画  
10 に同期して再生される音声（例えば、映画などにおいて、映画映像と対になって収録されている音声）とは別途に用意された音声データが、字幕やボタンの表示制御に同期して再生される音声をいう。

#### 発明の開示

15 この発明の目的は、大容量の再生専用光ディスクにおいて自由度の高いユーザインターフェイスを実現するための再生装置、再生方法、再生プログラムおよび記録媒体を提供することにある。

この発明の他の目的は、大容量の再生専用光ディスクにおいて、より表現豊かなユーザインターフェイスを実現する再生装置、再生方法  
20 、再生プログラムおよび記録媒体を提供することにある。

上述した課題を解決するために、第1の発明は、コンテンツデータを再生する再生装置において、少なくともプログラムコードおよびユーザに対して操作を促す操作画面を構成する画像データを含む非リアルタイムストリームと、少なくとも動画データおよび字幕データを含  
25 むリアルタイムストリームとが入力される入力手段と、入力手段により入力されたプログラムコードを格納するコード格納手段と、入力手

段により入力された画像データを格納する画像データ格納手段と、入力手段により入力された動画データをデコードした復号動画データと、入力手段により入力された字幕データをデコードした復号字幕データとを合成する第1の合成手段と、コード格納手段に格納されたプログラムコードに基づき、画像データ格納手段に格納された画像データをデコードした復号画像データと、第1の合成手段により合成された動画字幕合成データとを合成する第2の合成手段とを有することを特徴とする再生装置である。

第2の発明は、コンテンツデータを再生する再生方法において、少なくともプログラムコードおよびユーザに対して操作を促す操作画面を構成する画像データを含む非リアルタイムストリームと、少なくとも動画データおよび字幕データを含むリアルタイムストリームとを入力する入力のステップと、入力のステップにより入力されたプログラムコードをコード格納手段に格納するステップと、入力のステップにより入力された画像データを画像データ格納手段に格納するステップと、入力のステップにより入力された動画データをデコードした復号動画データと、入力のステップにより入力された字幕データをデコードした復号字幕データとを合成する第1の合成のステップと、コード格納手段に格納されたプログラムコードに基づき、画像データ格納手段に格納された画像データをデコードした復号画像データと、第1の合成のステップにより合成された動画字幕合成データとを合成する第2の合成のステップとを有することを特徴とする再生方法である。

第3の発明は、コンテンツデータを再生する再生方法をコンピュータ装置に実行させる再生プログラムにおいて、再生方法は、少なくともプログラムコードおよびユーザに対して操作を促す操作画面を構成する画像データを含む非リアルタイムストリームと、少なくとも動画



データおよび字幕データを含むリアルタイムストリームとを入力する  
入力のステップと、入力のステップにより入力されたプログラムコード  
をコード格納手段に格納するステップと、入力のステップにより入力  
された画像データを画像データ格納手段に格納するステップと、入  
5 力のステップにより入力された動画データをデコードした復号動画デ  
ータと、入力のステップにより入力された字幕データをデコードした  
復号字幕データとを合成する第1の合成のステップと、コード格納手  
段に格納されたプログラムコードに基づき、画像データ格納手段に格  
納された画像データをデコードした復号画像データと、第1の合成の  
10 ステップにより合成された動画字幕合成データとを合成する第2の合  
成のステップとを有することを特徴とする再生プログラムである。

第4の発明は、コンテンツデータを再生する再生方法をコンピュー  
タ装置に実行させる再生プログラムが記録された記録媒体において、  
再生方法は、少なくともプログラムコードおよびユーザに対して操作  
15 を促す操作画面を構成する画像データを含む非リアルタイムストリー  
ムと、少なくとも動画データおよび字幕データを含むリアルタイムス  
トリームとを入力する入力のステップと、入力のステップにより入力  
されたプログラムコードをコード格納手段に格納するステップと、入  
力のステップにより入力された画像データを画像データ格納手段に格  
20 納するステップと、入力のステップにより入力された動画データをデ  
コードした復号動画データと、入力のステップにより入力された字幕  
データをデコードした復号字幕データとを合成する第1の合成のステ  
ップと、コード格納手段に格納されたプログラムコードに基づき、画  
像データ格納手段に格納された画像データをデコードした復号画像デ  
25 ータと、第1の合成のステップにより合成された動画字幕合成データ  
とを合成する第2の合成のステップとを有することを特徴とする記録

媒体である。

第5の発明は、コンテンツデータが記録された円盤状の形状を有する記録媒体において、少なくともプログラムコードおよびユーザに対して操作を促す操作画面を構成する画像データを含む非リアルタイム  
5 ストリームと、少なくとも動画データおよび字幕データを含むリアルタイムストリームとが記録され、プログラムコードに基づき、再生後に画像データ格納手段に格納された画像データをデコードした復号画像データと、再生された動画データをデコードした復号動画データおよび再生された字幕データをデコードした復号字幕データが合成された動画字幕合成データとが合成されることを特徴とする記録媒体である。  
10

上述したように、第1、第2、第3および第4の発明は、少なくともプログラムコードおよびユーザに対して操作を促す操作画面を構成する画像データを含む非リアルタイムストリームと、少なくとも動画  
15 データおよび字幕データを含むリアルタイムストリームとを入力し、入力されたプログラムコードおよび画像データをコード格納手段および画像データ格納手段にそれぞれ格納し、入力された動画データおよび字幕データをそれぞれデコードした復号動画データおよび復号字幕データとを合成して動画字幕合成データとし、コード格納手段に格納  
20 されたプログラムコードに基づき、画像データ格納手段に格納された画像データをデコードした復号画像データと動画字幕合成データとを合成するようにしているため、再生時に、同一の画像データを用いた操作画面を、異なるタイミングでそれぞれ表示させることが容易である。

25 第5の発明は、少なくともプログラムコードおよびユーザに対して操作を促す操作画面を構成する画像データを含む非リアルタイムスト

リームと、少なくとも動画データおよび字幕データを含むリアルタイムストリームとが記録され、プログラムコードに基づき、再生後に画像データ格納手段に格納された画像データをデコードした復号画像データと、再生された動画データをデコードした復号動画データおよび再生された字幕データをデコードした復号字幕データが合成された動画字幕合成データとが合成されるようにしているため、再生時に、同一の画像データを用いた操作画面を、異なるタイミングでそれぞれ表示させることが容易である。

## 10 図面の簡単な説明

第1図は、従来技術による一例のDVDメニュー画面を示す略線図、第2図A、第2図Bおよび第2図Cは、リモコン操作により選択状態のボタンを変更する様子を示す略線図、第3図は、決定ボタンの操作によりボタンが実行状態を表す色に変更される様子を示す略線図、第4図A、第4図Bおよび第4図Cは、従来技術によるメニュー画面の構成の例を示す略線図、第5図は、従来技術によるサブピクチャの一例のデータ構成を示す略線図、第6図は、AVストリームファイルの再生順序指定の仕組みを模式的に示す略線図、第7図は、クリップAVストリーム、クリップ情報、クリップ、プレイアイテムおよびプレイリストの関係を示すUML図、第8図は、複数のプレイリストから同一のクリップを参照する方法を説明するための略線図、第9図は、記録媒体に記録されるファイルの管理構造を説明するための略線図、第10図は、ファイル「info.bdav」の一例の構造を表すシンタックスを示す略線図、第11図は、ブロックUIAppInfoBDAV()の一例の構造を表すシンタックスを示す略線図、第12図は、ブロックTableOfPlayLists()の一例の構造を表すシンタックスを示す略線図、第13図は、

ファイル「xxxxx.rpls」および「yyyyy.vpls」の一例の構造を表すシンタクスを示す略線図、第14図は、ブロックUIAppInfoPlayList()の一例の構造を表すシンタクスを示す略線図、第15図は、ブロックPlayList()の一例の構造を表すシンタクスを示す略線図、第16図は、  
5、ブロックPlayItem()の一例の構造を表すシンタクスを示す略線図、第17図は、ブリッジクリップを説明するための略線図、第18図は、ブロックPlayListMark()の一例の構造を表すシンタクスを示す略線図、第19図は、ファイル「zzzzz.clpi」の一例の構造を表すシンタクスを示す略線図、第20図は、この発明の実施の一形態で画像の表示系として用いられるプレーン構造の一例を示す略線図、第21図は、  
10、動画プレーン、字幕プレーンおよびグラフィクスプレーンの一例の解像度および表示可能色を示す略線図、第22図は、動画プレーン、字幕プレーンおよびグラフィクスプレーンを合成する一例の構成を示すブロック図、第23図は、パレットの入出力データの一例を示す略線図、第24図は、パレットに格納される一例のパレットテーブルを示す略線図、第25図は、グラフィクスプレーンに表示されるメニュー画面の一例を示す略線図、第26図は、シナリオの一例の内部構造を示す略線図、第27図は、シナリオの一例の内部構造を示す略線図、第28図は、シナリオの構成の分類を説明するための略線図、第2  
20 9図A、第29図Bおよび第29図Cは、プレイリストの構成の分類を説明するための略線図、第30図は、タイトルおよびチャプタについて説明するための略線図、第31図は、BD仮想プレーヤモデルについて説明するための略線図、第32図A、第32図Bおよび第32図Cは、BD仮想プレーヤにおいて独自に定義される一例のイベント  
25 を示す略線図、第33図A、第33図B、第33図C、第33図D、第33図E、第33図F、第33図Gおよび第33図Hは、この発明

の実施の一形態で定義される、BD仮想プレーヤが有するコマンドの一例を示す略線図、第34図A、第34図B、第34図C、第34図D、第34図E、第34図F、第34図G、第34図H、第34図Iおよび第34図Jは、この発明の実施の一形態で定義される、BD仮想プレーヤが有するコマンドの一例を示す略線図、第35図Aおよび第35図Bは、独自コマンドを記述言語として用いたシナリオで記述されるコマンドによるBD仮想プレーヤの動作を概略的に示すフローチャート、第36図Aおよび第36図Bは、プレイリストの再生動作を説明するためのフローチャート、第37図Aおよび第37図Bは、シナリオの一例の階層構造を示す略線図、第38図は、ディスクに記録されるシナリオ構成の一例を示す略線図、第39図は、シナリオを構成する際に必要とされる一例のファイルを一覧して示す略線図、第40図は、シナリオがディスクに記録される際の一例のディレクトリ構造を示す略線図、第41図は、スクリプトファイルのより具体的な記述の例を示す略線図、第42図は、スクリプトファイルのより具体的な記述の例を示す略線図、第43図は、HTMLファイルのより具体的な記述の例を示す略線図、第44図は、スクリプトファイルのより具体的な記述の例を示す略線図、第45図は、記録媒体に記録されるファイルの管理構造を説明するための略線図、第46図は、シナリオを記述するためのシナリオファイル(scenario.hdmv)の一例の構造を表すシンタクスを示す略線図、第47図は、ブロックAutoplay()の一例の構造を表すシンタクスを示す略線図、第48図は、ブロックScenario()の一例の構造を表すシンタクスを示す略線図、第49図は、ブロックentrylist.dataの一例の構造を表すシンタクスを示す略線図、第50図は、ブロックAppInfo()の一例の構造を表すシンタクスを示す略線図、第51図は、ブロックScenarioEntry()の一例の構造を

- 表すシンタクスを示す略線図、第 5 2 図は、ファイルxxxxx.mplsの一例の構造を表すシンタクスを示す略線図、第 5 3 図は、ブロックPLCpntrolInfo()の一例の構造を表すシンタクスを示す略線図、第 5 4 図は、フィールドPL\_Playback\_typeのの取り得る値と意味を示す略線図
- 5 、第 5 5 図は、フィールドPL\_random\_access\_modeの取り得る値と意味を示す略線図、第 5 6 図は、ブロックPlayList()の一例の構造を表すシンタクスを示す略線図、第 5 7 図は、ブロックPlayItem()の一例の構造を表すシンタクスを示す略線図、第 5 8 図は、フィールドPI\_random\_access\_modeの取り得る値と意味を示す略線図、第 5 9 図は、
- 10 フィールドstill\_modeの取り得る値と意味を示す略線図、第 6 0 図は、フィールドis\_seamless\_angle\_changeの取り得る値と意味を示す略線図、第 6 1 図は、ブロックSubPlayItem()の一例の構造を表すシンタクスを示す略線図、第 6 2 図は、フィールドis\_repeat\_flagの取り得る値と意味を示す略線図、第 6 3 図は、メインパスとサブパスの時間軸の関係を示す略線図、第 6 4 図は、ファイルzzzzz.clpiの一例の構造を表すシンタクスを示す略線図、第 6 5 図は、ブロックClipInfo()の一例の構造を表すシンタクスを示す略線図、第 6 6 図は、フィールドapplication\_typeの取り得る値と意味を示す略線図、第 6 7 図は、ブロックSequenceInfo()の一例の構造を表すシンタクスを示す略線
- 20 図、第 6 8 図は、ブロックProgramInfo()の一例の構造を表すシンタクスを示す略線図、第 6 9 図は、ブロックStreamCodingInfo()の一例の構造を表すシンタクスを示す略線図、第 7 0 図は、ブロックCPI()の一例の構造を表すシンタクスを示す略線図、第 7 1 図は、フィールドCPI\_typeの取り得る値と意味を示す略線図、第 7 2 図は、ブロックEP\_map\_for\_one\_stream\_PID()の一例の構造を表すシンタクスを示す略線図、第 7 3 図A、第 7 3 図Bおよび第 7 3 図Cは、プレーヤデコ

ーダの一例の構成を示す機能ブロック図、第 7 4 図は、ボタン表示の一例の状態遷移図、第 7 5 図は、オブジェクトの種類の分類を説明するための図、第 7 6 図 A、第 7 6 図 B および第 7 6 図 C は、この発明の実施の一形態によるグラフィクスオブジェクトのデータ構造の例を示す略線図、第 7 7 図は、グラフィクスオブジェクトが分割されて P E S パケットに格納される様子を示す略線図、第 7 8 図は、グラフィクスオブジェクトのデコードを行うグラフィクスオブジェクトデコーダモデルの一例の構成を示す機能ブロック図、第 7 9 図 A、第 7 9 図 B、第 7 9 図 C および第 7 9 図 D は、グラフィクスオブジェクト入力バッファ、P N G デコーダ、オブジェクトバッファおよびプレーンバッファにおける蓄積データ量の遷移の例を概略的に示す略線図、第 8 0 図は、プレーンへのデータ転送速度について説明するための図、第 8 1 図 A および第 8 1 図 B は、ウィンドウの定義について説明するための図、第 8 2 図 A および第 8 2 図 B は、ウィンドウの定義について説明するための図、第 8 3 図は、グラフィクスオブジェクトの一例の構造を表すシンタクスを示す略線図、第 8 4 図は、ブロック GlobalPaletteTable() の一例の構造を表すシンタクスを示す略線図、第 8 5 図 A および第 8 5 図 B は、命令グループ DispCmds(i) に置かれる表示制御命令の例を一覧して示す略線図、第 8 6 図 A および第 8 6 図 B は、命令 set\_display\_box(x1,y1,x2,y2) および命令 set\_clipping\_box(a1,b1,a2,b2) を説明するための略線図、第 8 7 図は、座標軸の定義を説明するための略線図、第 8 8 図は、命令グループ DispCmds(i) の記述とグラフィクスオブジェクトの表示変化の一例を示す略線図、第 8 9 図 A、第 8 9 図 B、第 8 9 図 C および第 8 9 図 D は、字幕表示が徐々に現れるフェードインの例を示す略線図、第 9 0 図 A および第 9 0 図 B は、字幕の P N G 画像がプレーン内を移動する例を示す略線図、第

9 1 図Aおよび第9 1 図Bは、字幕表示をスクロールさせる例を示す略線図、第9 2 図Aおよび第9 2 図Bは、PNG画像の一部を表示する枠を設定し、PNG画像上でこの枠を移動させつつ、プレーン上での表示位置も移動させる例を示す略線図、第9 3 図は、ボタン画像に

5 音声データを割り当てたグラフィクスオブジェクトの一例のデータ構造を示す略線図、第9 4 図は、グラフィクスオブジェクトに音声データを格納しない場合の一例のデータの管理構造を示す略線図、第9 5 図は、グラフィクスオブジェクトに音声データを格納する一例の方法を示す略線図、第9 6 図は、グラフィクスオブジェクトに音声データを

10 を格納しない場合の音声データの再生についてより具体的に説明するための略線図である。

#### 発明を実施するための最良の形態

この発明の実施の一形態について説明する。この発明の実施の一形態では、記録再生規格であるBlu-ray Disc規格(Blu-ray Disc Rewritable Format Ver1.0)を基に、再生専用ディスクで必要となるインタラクティブ機能や、マルチアングル機能などの機能を実現できるようにした。

以下の説明は、下記の構成に従い行う。

- 20 1. BD-REフォーマットの概要
2. BD-ROMフォーマットの概要
- 2-1. プレーンについて
- 2-2. メニュー画面
- 2-3. シナリオについて
- 25 2-4. シナリオの分類
- 2-5. 仮想プレーヤモデルについて



2-6. コマンドについて

2-7. コマンドの実行について

2-8. シンタクスについて

2-9. デコーダモデル

5 2-10. ボタンについて

2-11. グラフィックスの転送速度について

2-12. グラフィックスオブジェクトについて

2-13. 効果音について

2-14. その他

10

## 1. BD-REフォーマットの概要

15 先ず、理解を容易とするために、Blu-ray Disc に関し、"Blu-ray Disc Rewritable Format Ver1.0 part3 Audio Visual Specifications"で規定されている、コンテンツすなわちディスクに記録されたAV (Audio/Video) データの管理構造について説明する。以下では、この管理構造をBD-AVフォーマットと称する。

例えばMPEG (Moving Pictures Experts Group) ビデオやMPEGオーディオなどの符号化方式で符号化され、MPEG-2システムに従い多重化されたビットストリームは、クリップAVストリーム（  
20 またはAVストリーム）と称される。クリップAVストリームは、Blu-ray Disc に関する規格の一つである"Blu-ray Disc Rewritable Format Ver1.0 part2"で定義されたファイルシステムにより、ファイルとしてディスクに記録される。このファイルを、クリップAVストリームファイル（またはAVストリームファイル）と称す  
25 る。

クリップAVストリームファイルは、ファイルシステム上での管理

単位であり、ユーザにとって必ずしも分かりやすい管理単位であるとは限らない。ユーザの利便性を考えた場合、複数のクリップA Vストリームファイルに分割された映像コンテンツを一つにまとめて再生するための情報や、クリップA Vストリームファイルの一部だけを再生  
5 するための情報、特殊再生や頭出し再生を滑らかに行うための情報などをデータベースとしてディスクに記録しておく必要がある。Blu-ray Discに関する規格の一つである"Blu-ray Disc Rewritable Format Ver1.0 part3"で、このデータベースが規定される。

第6図は、A Vストリームファイルの一部または全部の範囲を指定  
10 して、必要な部分だけを並べて再生する再生順序指定の仕組みを模式的に示す。第6図において、プレイリスト(PlayList)は、A Vストリームファイルの一部または全部の範囲を指定して、必要な部分だけを再生する指示を与える。ユーザがコンテンツの再生を行う際には、このプレイリストの単位で選択する。プレイリストは、ユーザからみて  
15 ひとまとまりであり、再生が連続に行われることをユーザが暗黙に期待する、映像・音声の単位である。

最も簡単なプレイリストの構成は、記録が開始されてから記録が終了されるまでの一繋ぎりのA Vストリームファイルからなり、編集をしなければ、これが一つのプレイリストになる。

20 プレイリストは、再生するA Vストリームファイルの指定と、指定されたA Vストリームファイルの再生箇所を指定する再生開始点と再生終了点の集まりとから構成される。この再生開始点と再生終了点の情報を一組としたものは、プレイアイテム(PlayItem)と称される。プレイリストは、プレイアイテムの集合で構成される。プレイアイテム  
25 を再生するということは、そのプレイアイテムに参照されるA Vストリームファイルの一部分を再生するということになる。

クリップAVストリームは、上述したように、ビデオデータやオーディオデータがMPEG2 TS（トランスポートストリーム）の形式などに多重化されたビットストリームである。このクリップAVストリームに関する情報がクリップ情報(Clip Information)としてファイルに記録される。

- クリップAVストリームファイルと、対応するクリップ情報が記録されたクリップ情報ファイルとをひとまとまりのオブジェクトと見なし、クリップ(Clip)と称する。クリップは、クリップAVストリームとクリップ情報とから構成される、一つのオブジェクトである。
- 10 ファイルは、一般的に、バイト列として扱われる。クリップAVストリームファイルのコンテンツは、時間軸上に展開され、クリップ中のエン트리ポイントは、主に時間ベースで指定される。所定のクリップへのアクセスポイントのタイムスタンプが与えられた場合、クリップAVストリームファイルの中でデータの読み出しを開始すべきアドレス情報を見つけるために、クリップ情報ファイルを用いることができる。

1のディスク中に記録された全てのプレイリストおよびクリップは、ボリュームインフォメーション(Volume Information)で管理される。

- 20 第7図は、上述のようなクリップAVストリーム、クリップ情報(Stream Attributes)、クリップ、プレイアイテムおよびプレイリストの関係を示すUML(Unified Modeling Language)図である。プレイリストは、1または複数のプレイアイテムに対応付けられ、プレイアイテムは、1のクリップに対応付けられる。1のクリップに対して、
- 25 それぞれ開始点および／または終了点が異なる複数のプレイアイテムを対応付けることができる。1のクリップから1のクリップAVスト

リームファイルが参照される。1のクリップから1のクリップ情報ファイルが参照される。クリップAVストリームファイルとクリップ情報ファイルとは、1対1の対応関係を有する。このような構造を定義することにより、クリップAVストリームファイルを変更することなく、任意の部分だけを再生する、非破壊の再生順序指定を行うことが可能となる。

第8図のように、複数のプレイリストから同一のクリップを参照することもできる。第8図の例では、クリップ1が2つのプレイリスト2および3から参照されている。第8図において、クリップ1は、横方向が時間軸を表す。プレイリスト2により、コマーシャル区間b-cとシーンeとを含むクリップ1の区間a~fが参照される。プレイリスト3により、シーンeを含むクリップ1の区間d-gが参照される。プレイリスト2を指定することで、クリップ1の区間a~fを再生することができ、プレイリスト3を指定することで、クリップ1の区間d-gを再生することができる。

"Blu-ray Disc Rewritable Format Ver1.0 part3"で規定された、記録媒体に記録されるファイルの管理構造について、第9図を用いて説明する。ファイルは、ディレクトリ構造により階層的に管理される。記録媒体上には、先ず、1つのディレクトリ（第9図の例ではルートディレクトリ）が作成される。このディレクトリの下が、1つの記録再生システムで管理される範囲とする。

ルートディレクトリの配下に、ディレクトリBD AVが置かれる。ディレクトリBD AVは、第9図のように、ディレクトリBD AV、BD AV 1、BD AV 2、・・・、BD AV nの如く、ルートディレクトリの配下に複数を置くことができる。以下では、複数のディレクトリBD AV、BD AV 1、BD AV 2、・・・、BD AV nをディ

レクトリBDVで代表させて説明する。

ディレクトリBDVの配下には、以下の6種類のファイルが置かれる。

(1) info.bdav

5 (2) menu.tidx、mark.tidx

(3) menu.tdt1、menu.tdt2、mark.tdt1、mark.tdt2

(4) #####.rpls、#####.vpls

(5) %%%%.clpi

(6) \*\*\*\*\*.m2ts

10 (4) のファイル「#####.rpls」および「#####.vpls」において、  
「#####」は、任意の番号を示す。(5) のファイル「%%%.clpi」  
において、「%%%」は任意の番号を示す。(6) のファイル「\*\*\*\*\*  
.m2ts」において、「\*\*\*\*\*」は、ファイル「\*\*\*\*\*.m2ts」がファイル  
「%%%.clpi」と一対一で対応するような番号である。番号「\*\*\*\*\*  
15 」は、番号「%%%」と同一とすることができる。

(1) のファイル「info.bdav」は、ディレクトリBDV全体の  
情報が格納されるファイルである。(2) のファイル「menu.tidx」  
および「mark.tidx」は、サムネイル画像の情報が格納されるファイル  
である。(3) のファイル「menu.tdt1」、「menu.tdt2」、「mark  
20 .tdt1」および「mark.tdt2」は、サムネイル画像が格納されるファイル  
である。各ファイルの拡張子「tdt1」および「tdt2」は、当該ファイル  
に格納されているサムネイル画像データが暗号化されているか否  
かを示す。

(4) のファイル「#####.rpls」、「#####.vpls」は、プレイリス  
25 トの情報が格納されるファイルである。ファイル「#####.rpls」、「  
#####.vpls」は、ディレクトリBDVの配下に設けられたディレク

トリPLAYLISTのさらに配下に置かれている。

(5) のファイル「%%%%.clpi」は、クリップ情報が格納されるファイルである。ファイル「%%%%.clpi」は、ディレクトリBD AVの配下に設けられたディレクトリCLIPINFのさらに配下に置かれている。 (6) のファイル「\*\*\*\*\*.m2ts」は、クリップAVストリームが格納される、クリップAVストリームファイルである。このクリップAVストリームファイルは、ファイル名の番号「\*\*\*\*\*」により、1つのクリップ情報ファイル「%%%%.clpi」に対応付けられる。ファイル「\*\*\*\*\*.m2ts」は、ディレクトリBD AVの配下に設けられたディレクトリSTREAMのさらに配下に置かれる。

各ファイルについて、より詳細に説明する。(1) のファイル「info.bdav」は、ディレクトリBD AVの配下に唯一、設けられる。第10図は、ファイル「info.bdav」の一例の構造を表すシンタクスを示す。シンタクスをコンピュータ装置などのプログラムの記述言語として用いられるC言語の記述法に基づき示す。他のシンタクスを表す図において、同様である。

第10図において、ファイル「info.bdav」の内部は、機能別の情報毎にブロックが構成される。フィールドtype\_indicatorには、文字列「BD AV」が格納され、このファイルがファイル「info.bdav」であることが示される。フィールドversion\_numberには、このファイル「info.bdav」のバージョンが示される。ブロックUIAppInfoBD AV()には、ディレクトリBD AVの配下に関する情報が格納される。ブロックTableOfPlayLists()には、プレイリストの並び順に関する情報が格納される。ブロックMakersPrivateData()には、記録再生装置のメーカー固有の情報が格納される。

ファイル「info.bdav」の先頭には、各ブロックの先頭を表すアド

レスが記述される。例えば、フィールドTableOfPlayLists\_Start\_addressには、ブロック「TableOfPlayLists()」の開始する位置がファイル内の相対バイト数で示される。

第11図は、ブロックUIAppInfoBDAV()の一例の構造を表すシンタ  
クスを示す。フィールドlengthには、フィールドlength直後のフィールドからこのブロックUIAppInfoBDAV()の最後まで  
の長さがバイトで示される。フィールドBDAV\_character\_setには、このブロックUIAppInfoBDAV()内のフィールドBDAV\_nameに記録されている文字列の文字セットが示される。文字セットとしては、ASCII、Unicode  
などが選択できる。

フラグBDAV\_protect\_flagは、ディレクトリBDAVの配下に置かれるコンテンツを、ユーザに制限無しに見せて良いか否かを示す。例えば、このフラグが値「1」にセットされており、且つ、ユーザが正しく暗証番号PIN(Personal Identification Number)を入力できた場合に、ディレクトリBDAVの配下に置かれたコンテンツをユーザに見せることができるものとする。一方、フラグBDAV\_protect\_flagが値「0」にセットされている場合には、ユーザによる暗証番号PINの入力が無くても、ディレクトリBDAVの配下に置かれたコンテンツをユーザに見せることができる。

暗証番号PINは、フィールドPINに記録される。暗証番号PINは、例えば4桁の0～9の数字からなり、上述のように再生制御を有効にした場合に必要となる暗証番号を表す。暗証番号PINのそれぞれの数字は、例えばISO(International Organization for Standardization)／IEC(International Electrotechnical Commission)646の規定に従って符号化される。

このブロックUIAppInfoBDAV()に示される以上の情報により、ディ

レクトリBD A Vに対する再生制限が規定される。なお、詳細は後述するが、個々のプレイリストに対する再生制限については、ファイル「#####.rpls」、「#####.vpls」内に示されるブロックUIAppInfoPlaylist()の中で定義されるフラグplayback\_control\_flagによって規定  
5 される。

この例では、ディレクトリBD A Vの配下のコンテンツの再生を再開する際に、優先して再生すべきプレイリストを指定する、レジューム機能を用いることができる。このレジューム機能は、前回視聴を中断した箇所から再生を再開したい場合などに使うことを想定している

10 。

第11図において、フラグresume\_valid\_flagは、このレジューム機能の有効／無効を示す。例えば、このフラグが値「0」にセットされている場合には、レジューム機能は無効とされる。このフラグが値「1」にセットされている場合には、レジューム機能が有効とされ、  
15 フィールドresume\_PlayList\_file\_nameで指定されるプレイリストを、優先して再生すべきプレイリストとして扱う。

フィールドref\_to\_menu\_thumbnail\_indexは、このディレクトリBD A Vを代表するサムネイル画像が存在する場合、そのサムネイル画像を特定するサムネイル番号が格納される領域である。Blu-ray Discの規格では、ディレクトリBD A Vを代表する静止画を特にメニューサムネイルと呼ぶ。フィールドref\_to\_menu\_thumbnail\_indexで指定されるインデックスthumbnail\_indexを持つサムネイル画像がこのディレクトリBD A Vのメニューサムネイルとなる。

フィールドBD A V\_name\_lengthは、フィールドBD A V\_nameに示される  
25 ディレクトリBD A V名のバイト長を示す。このフィールドBD A V\_nameにおいて、左から、フィールドBD A V\_name\_lengthで示されるだけの



バイト数が有効な文字列であり、それがこのディレクトリBD A Vの名前を示す。なお、フィールドBD A V\_nameの中で、フィールドBD A V\_name\_lengthで示された有効な文字列の後のバイト列には、どのような値が入っていても良い。

- 5     第12図は、ブロックTableOfPlayLists()の一例の構造を表すシンタクスを示す。フィールドnumber\_of\_PlayListsには、このディレクトリBD A Vの配下に置かれたプレイリストの数が示される。このプレイリスト数をループ変数とする、次に続く「for」文のループにおいてフィールドPlayList\_file\_nameにより示されるプレイリストの並びが、プレイリスト一覧表示画面などに表示される際のプレイリストの順序を示す。プレイリストは、フィールドPlayList\_file\_nameにより、「#####.rpls」や「#####.vpls」といったファイル名で指定される。

- 15     ファイル「#####.rpls」および「#####.vpls」は、上述したように、ディレクトリPLAYLISTの配下に置かれる。これらのファイルは、1つのプレイリストと一対一に対応している。

- 20     第13図は、ファイル「#####.rpls」および「#####.vpls」の一例の構造を表すシンタクスを示す。第13図において、ファイル「#####.rpls」および「#####.vpls」の内部は、機能別の情報毎にブロックが構成される。フィールドtype\_indicatorにこのファイルを示す文字列が格納され、フィールドversion\_numberにこのファイルのバージョンが示される。

- 25     ブロックUIAppInfoPlayList()には、このプレイリストに関する属性情報が格納される。ブロックPlayList()には、このプレイリストを構成するプレイアイテムに関する情報が格納される。ブロックPlayListMark()には、このプレイリストに付されるマークの情報が格納され

る。ブロックMakersPrivateData()には、このプレイリストファイルを記録した装置の、メーカー固有の情報が格納される。ファイル「####  
#.rpls」および「#####.vpls」の先頭に設けられるフィールドPlayLi  
st\_start\_address、PlaylistMark\_start\_addressおよびMakersPrivat  
5 eData\_start\_addressには、それぞれ対応するブロックの先頭アドレ  
スが例えば32ビットのアドレス情報として示される。

各ブロックの先頭アドレスがファイル「#####.rpls」および「####  
#.vpls」の先頭に示されるために、ブロックの前および／または後ろ  
に、任意の長さのデータpadding\_wordを挿入することができる。但し  
10 、ファイル「#####.rpls」および「#####.vpls」の最初のブロックで  
あるブロックUIAppInfoPlayList()の開始位置は、ファイルの先頭か  
ら320バイト目に固定される。

第14図は、ブロックUIAppInfoPlayList()の一例の構造を表すシ  
ンタクスを示す。ブロックUIAppInfoPlayList()は、プレイリストの  
15 再生において直接的に必要とされない、プレイリストに関する各種の  
属性情報が格納される。フィールドPlaylist\_character\_setは、プレ  
イリストに関する文字列情報の文字セットが指定される。

フラグplayback\_control\_flagには、情報表示やプレイリストの再  
生を暗証番号PINに基づき制限するか否かが指定される。例えばフラ  
20 グplayback\_control\_flagが値「1」であれば、ユーザにより正しい  
暗証番号PINが入力されなければ、プレイリストのサムネイル画像な  
どの情報表示や、プレイリストの再生ができないとされる。フラグwr  
ite\_protect\_flagは、消去禁止フラグである。例えばこのフラグwrit  
e\_protect\_flagが値「1」であれば、ユーザが容易にプレイリストを  
25 消去できないようなユーザインタフェースにする必要がある。フラグ  
is\_played\_flagは、このプレイリストが再生済みであることを示す。

フラグis\_edited\_flagは、このプレイリストが編集されたことを示す。

フィールドtime\_zoneは、このプレイリストが記録された地点のタイムゾーンを示す。フィールドrecord\_time\_and\_dateは、プレイリス  
5 トの記録日時を示す。フィールドPlayList\_durationは、プレイリス  
トの再生時間を示す。

フィールドmaker\_IDおよびmaker\_model\_codeには、このプレイリス  
トを最終更新した記録機のメーカーおよび記録機をそれぞれ特定する情  
報が例えば番号で示される。フィールドchannel\_numberには、記録し  
10 たクリップA Vストリームのチャンネル番号が示され、フィールドch  
annel\_nameには、当該チャンネル名が示される。フィールドchannel\_  
nameに示されるチャンネル名の長さがフィールドchannel\_name\_lengt  
hに示される。フィールドchannel\_name の領域のうち、フィールドch  
annel\_name\_lengthで示される長さの文字列が有効となる。フィール  
15 ドPlayList\_nameには、フィールドPlayList\_name\_lengthに示される  
値を有効長としてプレイリスト名が示される。フィールドPlayList\_d  
etailには、フィールドPlayList\_detail\_lengthに示される値を有効  
長としてプレイリストの詳細情報が示される。

第15図は、ブロックPlayList()の一例の構造を表すシンタクスを  
20 示す。フィールドlengthは、フィールドlengthの直後のフィールドか  
らこのブロックPlayList()の終端までのバイト長を示す。フィールド  
PL\_CPI\_typeは、このプレイリストが持つC P I (Characteristic Po  
int Information: 特徴点情報)の種類を示す。フィールドnumber\_of  
\_PlayItemsは、このプレイリストを構成するプレイアイテムの数を示  
25 す。フィールドnumber\_of\_SubPlayItemsは、このプレイリストに付け  
られているアフレコオーディオ用のプレイアイテム (サブプレイアイ

テム) の数を示す。詳細は省略するが、プレイリストは、所定の条件を満たす場合にだけ、サブプレイアイテムを持つことができる。

ブロックPlayItem() は、プレイアイテムの情報が格納される。ブロックSubPlayItem() は、サブプレイアイテムの情報が格納される。

- 5     第16図は、ブロックPlayItem() の一例の構造を表すシンタクスを示す。フィールドClip\_Information\_file\_nameには、このプレイアイテムが参照しているクリップと一対一に対応するクリップ情報ファイル(拡張子がclpiであるファイル)のファイル名が文字列で示される。クリップ情報ファイルは、拡張子が「c l p i」とされているファイルである。

- 10     フィールドClip\_codec\_identifierは、このプレイアイテムにより参照されるクリップの符号化方式を示す。この例では、フィールドClip\_codec\_Identifierは、値「M2TS」に固定的とされる。フィールドconnection\_conditionは、このプレイアイテムが次のプレイアイテムとの間でどのような接続がされているかを示す情報である。フィールドconnection\_conditionにより、プレイアイテムとプレイアイテムとの間が継ぎ目なくシームレスに再生できるか否かが示される。

- 15     フィールドref\_to\_STC\_idは、このプレイアイテムにより参照されるクリップ内のシーケンスSTC\_sequenceを指定する。シーケンスSTC\_sequenceは、MPEG2 TS (Transport Stream)における時間軸の基準であるPCR (Program Clock Reference)が連続な範囲を表すBlu-ray Disc規格独自の構造である。シーケンスSTC\_sequenceには、クリップ内で一意な番号STC\_idが割り当てられる。このシーケンスSTC\_sequence内では、不連続の無い一貫した時間軸を定義できる
- 20     ので、プレイアイテムの開始時刻および終了時刻を一意に定めることができる。各プレイアイテムの開始点と終了点は、同一のシーケ

ンスSTC\_sequenceに存在していなければならない。フィールドref\_to\_STC\_idでは、番号STC\_idによりシーケンスSTC\_sequenceが指定される。

フィールドIN\_timeおよびOUT\_Timeは、このプレイアイテムにおける開始点および終了点の、シーケンスSTC\_sequence 上でのタイムスタンプpts (presentation\_time\_stamp)をそれぞれ示す。

ブロックBridgeSequenceInfo()には、ブリッジクリップ(Bridge-Clip)に関する情報が格納される。ブリッジクリップは、第17図に一例が示されるように、プレイアイテム間を継ぎ目無く、シームレスに再生する機能を実現する際に作成されるビットストリームである。前のプレイアイテムと現在のプレイアイテムとの継ぎ目の部分で、本来再生すべきビットストリームとは異なるビットストリーム、すなわちブリッジクリップを代わりに再生することで、2つのプレイアイテムをシームレスに再生することができる。

第18図は、ブロックPlayListMark()の一例の構造を表すシンタクスを示す。ブロックPlayListMark()は、マークの情報が格納されるデータ構造である。マークは、プレイリスト上の時刻を保持するための構造であって、マークを用いることにより、プレイリスト中への頭出し点の設定、プレイリストを細分化するチャプタの構造を設ける、などの機能が実現される。後述するグラフィクスプレーン上に描画された画像の表示開始/停止のタイミングを、マークを用いて指定することができる。

フィールドlengthは、フィールドlengthの直後のフィールドからブロックPlayListMark()の終端までのバイト長を示す。フィールドnumber\_of\_PlayList\_marksは、プレイリスト中のマークの数を表す。「for」文での1回のループが1つのマークの情報を示す。フラグmark\_in

valid\_flagは、このマークが有効であるか否かを示す。例えば、フラグmark\_invalid\_flagが値「0」でそのマークが有効であることが示され、値「1」であれば、そのマークの情報がデータベース上に存在するが、当該マークがユーザには見えない無効マークとされていることを示す。

フィールドmark\_typeは、このマークの種類を表す。マークの種類としては、プレイリストのサムネイル画像（代表画）とする映像の位置を示すマーク、どこまで再生したかを示すレジュームマーク、頭出し点を示すチャプタマーク、ある区間の再生を飛ばして先に進むことを示すスキップマーク、グラフィクス画像の読み込み開始タイミングを示すマーク、グラフィクス画像の表示開始のタイミングを示すマーク、グラフィクス画像の表示停止のタイミングを示すマークなどがある。

フィールドmark\_name\_lengthは、後述するフィールドmark\_nameのデータ長を示す。フィールドmaker\_IDは、このマークを作成した記録機の製造者を示す。これは、製造者独自のマークを識別する際に用いられる。フィールドref\_to\_PlayItem\_idには、このマークにより指示される時刻がどのプレイアイテム上に存在するかが示される。フィールドmark\_time\_stampは、このマークにより指示される時刻が示される。

フィールドentry\_ES\_PIDには、このマークがどのエレメンタリストリームすなわち映像データおよび／または音声データが符号化されたストリームに対して付与されたものかが示される。フィールドref\_to\_menu\_thumbnail\_indexおよびフィールドref\_to\_mark\_thumbnail\_indexは、マークを視覚的に表すサムネイル画像を示す。サムネイル画像としては、例えばそのマークにより指定される時刻の映像を静止画と

して抜き出した画像が考えられる。

フィールドdurationには、マークが時間軸上で長さを持つ場合に使われる。例えばスキップマークであれば、どのくらいの時間ジャンプをするかがフィールドdurationにより指定される。

- 5     フィールドmakers\_informationは、製造者独自の情報を記録するための領域である。フィールドmark\_nameには、マークに名前をつけた際、その名前を格納するための領域であり、サイズが上述のフィールドmark\_name\_lengthで指定される。

- 第19図は、ファイル「%%%%.clpi」の一例の構造を表すシンタックスを示す。上述したように、ファイル「%%%%.clpi」は、ディレクトリCLIPINFの配下に置かれ、各AVストリームファイル（ファイル「\*\*\*\*\*.m2ts」）毎に作られる。ファイル「%%%%.clpi」の内部は、機能別の情報毎にブロックが構成される。フィールドtype\_indicatorにこのファイルを示す文字列が格納され、フィールドversion\_numberにこのファイルのバージョンが示される。
- 10     フィールドtype\_indicatorにこのファイルを示す文字列が格納され、フィールドversion\_numberにこのファイルのバージョンが示される。
- 15

- ブロックClipInfo()には、クリップに関する情報が格納される。ブロックSequenceInfo()には、MPEG2システムにおけるトランスポートストリームの時刻基準を表すPCRの不連続点に関する情報が格納される。ブロックProgramInfo()には、MPEG2システムのプログラムに関する情報が格納される。ブロックCPI()には、ランダムアクセス開始可能点などの、AVストリーム中の特徴的な箇所を表す特徴点情報CPIに関する情報が格納される。ブロックClipMark()には、クリップに付された頭出しのためのインデックス点やコマーシャルの開始および／または終了点などのマーク情報が格納される。ブロックMakersPrivateData()には、記録機の製造者独自の情報が格納される。
- 20     プログラムに関する情報が格納される。ブロックCPI()には、ランダムアクセス開始可能点などの、AVストリーム中の特徴的な箇所を表す特徴点情報CPIに関する情報が格納される。ブロックClipMark()には、クリップに付された頭出しのためのインデックス点やコマーシャルの開始および／または終了点などのマーク情報が格納される。ブロックMakersPrivateData()には、記録機の製造者独自の情報が格納される。
- 25

これら各ブロックのファイル「%%.clpi」中での先頭を表すアドレス情報が、当該ファイルの先頭部分にフィールドSequenceInfo\_start\_address、ProgramInfo\_start\_address、CPI\_start\_address、ClipMark\_start\_addressおよびMakersPrivateData\_start\_addressとして

5 示される。

BDAVフォーマットは、以上に概略的に説明したようなデータ構造を持つことにより、クリップAVストリーム中の再生したい部分を開始および終了点の組で指定したプレイアイテムの並びでプレイリストを構築し、ユーザが認識するひとまとまりの再生単位を管理すること

10 が可能となる。

## 2. BDAVフォーマットの概要説明

次に、この発明の実施の一形態について説明する。この発明では、上述したBDAVフォーマットを拡張し、再生専用ディスク（BD-ROM：Blu-ray Disc-Read Only Memory）に適したフォーマットを

15 構築する。拡張されたBDAVフォーマットを、BDMVフォーマットと称する。

ディスクの内容を提示するメニュー画面を実現するためのプレーン構成を説明する。コンテンツ制作者側がプレイリストの再生順序を指定できるようにするためのシナリオ構造を追加する。そのシナリオ構造において、再生専用ディスクに特徴的なスチル（一時停止）、ランダム・シャッフル再生、マルチアングルなどの機能を実現するために必要なデータと、その格納方法について説明する。

### 25 2-1. プレーンについて

この発明の実施の一形態では、画像の表示系について、第20図に



一例が示されるようなプレーン構成を取る。動画プレーン10は、最も後ろ側（ボトム）に表示され、プレイリストで指定された画像（主に動画データ）が扱われる。字幕プレーン11は、動画プレーン10の上に表示され、動画再生中に表示される字幕データが扱われる。グラフィクスプレーン12は、最も前面に表示され、メニュー画面を表示するための文字データやボタンを表すビットマップデータなどのグラフィクスデータが扱われる。1つの表示画面は、これら3つのプレーンが合成されて表示される。

従来のDVDビデオと異なる特徴的な点は、従来の技術で説明した字幕やメニュー画面、ボタンなどが表示されるサブピクチャを、字幕プレーン11およびグラフィクスプレーン12とに分離し、字幕とボタンとをそれぞれ独立的に制御できるようにしたことである。従来のDVDビデオでは、メニュー画面やボタンなどのグラフィクス表示と、字幕表示とを、同じ仕組みで制御しており、これらを同一プレーン上に表示していた。同時に表示できるビットマップ画像は、1枚に限られていた。そのため、DVDビデオでは、複数のビットマップ画像を同時に表示することができなかった。この発明では、字幕を表示する字幕プレーン11と、グラフィクス表示を行うグラフィクスプレーン12とをそれぞれ独立的に設けることで、DVDビデオの持つ問題点を解消している。

字幕プレーン11およびグラフィクスプレーン12が、"Blu-ray Disc Rewritable Format Ver1.0 part3"に対する拡張部分であると考えることができる。

動画プレーン10、字幕プレーン11およびグラフィクスプレーン12は、それぞれ独立して表示が可能とされ、第21図に一例が示されるような解像度および表示可能色を有する。動画プレーン10は、

解像度が1920画素×1080ラインで1画素あたりに換算したデータ長が16ビットであって、輝度信号Y、色差信号Cb、Crが4:2:2のシステム（以下、YCbCr（4:2:2））とされる。YCbCr（4:2:2）は、各画素あたり輝度信号Yが8ビット、  
5 色差信号Cb、Crがそれぞれ8ビットで、色差信号Cb、Crが水平2画素で一つの色データを構成すると見なすカラーシステムである。

字幕プレーン11は、1920画素×1080ラインで各画素のサンプリング深さが8ビットとされ、カラーシステムは、256色のパ  
10 レットを用いた8ビットカラーマップアドレスとされる。

グラフィクスプレーン12は、解像度が1920画素×1080ラインで各画素のサンプリング深さが8ビットとされ、カラーシステムは、256色のパレットを用いた8ビットカラーマップアドレスとされる。

15 上述では、字幕プレーン11とグラフィクスプレーン12のカラーシステムを、256色のパレットを用いた8ビットカラーマップアドレスとしたが、この例に限定されない。色数については、サンプリング深さを変えてパレットの色数を増やせばよい。例えばサンプリング深さを12ビットとすれば、パレットで使用可能な色数を4096色  
20 とすることができる。サンプリング深さを24ビットとして、パレットを持たずに各画素が色情報を持つようにしたYCbCr（4:4:4）およびRGB（4:4:4）も、同様の仕組みで可能である。

グラフィクスプレーン12および字幕プレーン11は、256段階のアルファブレンディングが可能とされており、他のプレーンとの合  
25 成の際に、透明度を256段階で設定することが可能とされている。透明度の設定は、画素毎に行うことができる。以下では、透明度 $\alpha$ が

( $0 \leq \alpha \leq 1$ ) の範囲で表され、透明度  $\alpha = 0$  で完全に透明、透明度  $\alpha = 1$  で完全に不透明であるものとする。

字幕プレーン 11 では、例えば PNG (Portable Network Graphics) 形式の画像データが扱われる。グラフィクスプレーン 12 でも、PNG 形式の画像データを扱うことができる。PNG 形式は、1 画素のサンプリング深さが 1 ビット～16 ビットとされ、サンプリング深さが 8 ビットまたは 16 ビットの場合に、アルファチャンネル、すなわち、それぞれの画素成分の透明度情報（アルファデータと称する）を付加することができる。サンプリング深さが 8 ビットの場合には、256 段階で透明度を指定することができる。このアルファチャンネルによる透明度情報を用いてアルファブレンディングが行われる。256 色までのパレットイメージを用いることができ、予め用意されたパレットの何番目の要素（インデックス）であるかがインデックス番号により表現される。

字幕プレーン 11 およびグラフィクスプレーン 12 で扱われる画像データは、PNG 形式に限定されない。JPEG 方式など他の圧縮符号化方式で圧縮符号化された画像データや、ランレングス圧縮された画像データ、圧縮符号化がなされていないビットマップデータなどを扱うようにしてもよい。

第 22 図は、上述の第 20 図および第 21 図に従い 3 つのプレーンを合成する一例の構成を示す。動画プレーン 10 の動画データが 422/444 変換回路 20 に供給される。動画データは、422/444 変換回路 20 でカラーシステムが YCbCr (4:2:2) から YCbCr (4:4:4) に変換され、乗算器 21 に入力される。422/444 変換回路 20 と乗算器 21 との間に解像度変換回路を挿入し、動画データの解像度を変換するようにしてもよい。

字幕プレーン 1 1 の画像データがパレット 2 2 に入力され、RGB (4 : 4 : 4) の画像データとして出力される。この画像データに対してアルファブレンディングによる透明度が指定されている場合には、指定された透明度  $\alpha 1$  ( $0 \leq \alpha 1 \leq 1$ ) がパレット 2 2 から出力される。

第 2 3 図は、パレット 2 2 の入出力データの一例を示す。パレット 2 2 には、例えば PNG 形式のファイルに対応したパレット情報がテーブルとして格納される。パレット 2 2 からは、入力された 8 ビットの画素データをアドレスとして、インデックス番号が参照される。このインデックス番号に基づき、それぞれ 8 ビットのデータからなる RGB (4 : 4 : 4) のデータが出力される。それと共に、パレット 2 2 からは、透明度を表すアルファチャンネルのデータが取り出される。

第 2 4 図は、パレット 2 2 に格納される一例のパレットテーブルを示す。256 個のカラーインデックス値  $[0x00] \sim [0xFF]$  ( $[0x]$  は 16 進表記であることを示す) のそれぞれに対して、各々 8 ビットで表現される三原色の値 R、G および B と、透明度  $\alpha$  とが割り当てられる。パレット 2 2 は、入力された PNG 形式の画像データに基づきパレットテーブルが参照され、画像データにより指定されたインデックス値に対応する、それぞれ 8 ビットのデータからなる R、G および B 各色のデータ (RGB データ) と、透明度  $\alpha$  とを画素毎に出力する。後述するパレット 2 6 にも、同様のパレットテーブルが格納される。

パレット 2 2 から出力された RGB データは、RGB/YCbCr 変換回路 2 9 に供給され、各データ長が 8 ビットの輝度信号 Y と色信号 Cb、Cr のデータに変換される (以下、まとめて YCbCr データ

タと称する)。これは、以降のプレーン間合成を共通のデータ形式で行う必要があるため、動画データのデータ形式であるYCbCrデータに統一している。

RGB/YCbCr変換回路29から出力されたYCbCrデータ  
5 および透明度データ $\alpha 1$ とがそれぞれ乗算器23に入力される。RGB/YCbCr変換回路29と乗算器23との間に解像度変換回路を挿入し、YCbCrデータの解像度を変換するようにしてもよい。乗算器23では、入力されたYCbCrデータに透明度データ $\alpha 1$ が乗  
10 ぜられる。乗算結果は、加算器24の一方の入力端に入力される。乗算器23では、YCbCrデータにおける輝度信号Y、色差信号Cb、Crのそれぞれについて、透明度データ $\alpha 1$ との乗算が行われる。透明度データ $\alpha 1$ の補数 $(1 - \alpha 1)$ が乗算器21に供給される。

乗算器21では、422/444変換回路20から入力された動画データに透明度データ $\alpha 1$ の補数 $(1 - \alpha 1)$ が乗ぜられる。乗算結  
15 果は、加算器24の他方の入力端に入力される。加算器24において、乗算器21および23の乗算結果が加算される。これにより、動画プレーン10と字幕プレーン11とが合成される。加算器24の加算結果が乗算器25に入力される。

グラフィクスプレーン12の画像データも字幕プレーン11と同様  
20 に、パレットテーブル26によりRGB(4:4:4)のデータが出力され、RGB/YCbCr変換回路27に入力される。グラフィクスプレーン12の画像データのカラーシステムがRGB(4:4:4)である場合には、カラーシステムがYCbCr(4:4:4)に変換されてRGB/YCbCr変換回路27から出力される。RGB/YCbCr変換回路27から出力されたYCbCrデータが乗算器2  
25 8に入力される。RGB/YCbCr変換回路27と乗算器28との

間に解像度変換回路を挿入し、Y C b C r データの解像度を変換するようにしてもよい。

パレット 2 6 において、インデックス値に対してアルファブレンディングによる透明度が指定されている場合には、指定された透明度  $\alpha_2$  (  $0 \leq \alpha_2 \leq 1$  ) がパレット 2 6 から出力される。透明度データ  $\alpha_2$  は、乗算器 2 8 に供給される。乗算器 2 8 では、R G B / Y C b C r 変換回路 2 7 から入力された Y C b C r データに対し、輝度信号 Y、色差信号 C b、C r のそれぞれについて、透明度データ  $\alpha_2$  との乗算が行われる。乗算器 2 8 による乗算結果が加算器 2 9 の一方の入力端に入力される。透明度データ  $\alpha_2$  の補数 (  $1 - \alpha_2$  ) が乗算器 2 5 に供給される。

乗算器 2 5 では、加算器 2 4 の加算結果に対して透明度データ  $\alpha_2$  の補数 (  $1 - \alpha_2$  ) が乗ぜられる。乗算器 2 5 の乗算結果は、加算器 2 7 の他方の入力端に入力され、上述した乗算器 2 8 による乗算結果と加算される。これにより、動画プレーン 1 0 と字幕プレーン 1 1 との合成結果に対して、グラフィクスプレーン 1 2 が合成される。

字幕プレーン 1 1 およびグラフィクスプレーン 1 2 において、例えば、表示すべき画像の無い領域の透明度  $\alpha = 0$  と設定することで、そのプレーンの下に表示されるプレーンを透過表示させることができる。そうすることにより、動画プレーン 1 0 に表示されている動画データを、字幕プレーン 1 1 やグラフィクスプレーン 1 2 の背景として表示することができる。

なお、この第 2 2 図に示される構成は、ハードウェアおよびソフトウェアの何れでも実現可能なものである。

上述では、グラフィクスプレーン 1 2 の解像度を 1 9 2 0 画素  $\times$  1 0 8 0 ライン、カラーシステムを 2 5 6 色のカラーパレットを用いた

8ビットカラーマップアドレスとしたが、グラフィクスプレーン12の解像度および色数は、上述に限定されない。

例えば、解像度を960画素×540ラインとし、サンプリング深さを24ビットとして各画素にRGB各色にそれぞれ8ビットの色情報を持たせ、さらに各画素に8ビットのアルファデータを持たせるようにできる。使用可能な色数が上述の256色に比べて格段に増大し、表現力が増す。画素数が減るのでプレーンの書き換え速度が低下することもない。グラフィクスプレーン12の用途を、将来的に自然画や高速のアニメーションの描画に拡張した際に、有用であると考えられる。

この例では、上述の1920画素×1080ラインの例に対して全体の画素数が1/4になる一方、各画素毎のデータ量が8ビットから32ビットと4倍になるので、グラフィクスプレーン12全体のデータ量に変化がない。そのため、追加のメモリは必要なく、フレームメモリの使用方法を変えるだけで容易に実現可能である。

また、RGB各色にそれぞれ8ビットずつの色情報を持たせており、表示可能色数が十分とされるため、第22図におけるパレット22を省略できる。図示は省略するが、グラフィクスプレーン12の画像データは、直接的にRGB/YCbCr変換回路27に入力されることになる。アルファデータ $\alpha$ 2は、RGB/YCbCr変換回路27の前で抜き出されて、乗算器28に供給される。

グラフィクスプレーン12と合成される動画プレーン10および字幕プレーン11の解像度は、1920画素×1080ラインのままである。実際の表示においては、動画プレーン10または字幕プレーン11の例えば2画素×2ラインからなる4画素に対して、解像度を1/4にしたグラフィクスプレーン12の1画素を繰り返し表示させ、

見かけ上の解像度を合わせた後に、動画プレーン10および字幕プレーン11との合成を行う。

5 以上のようなプレーンを設定することで、再生専用規格に必要なメニュー画面とボタンの表示を可能とする。メニュー画面上のボタンが選択されると、そのボタンに対応したプレイリストが再生されることになる。このとき、プレイリスト間がどのように結び付けられているかという情報がディスク上に記録されていることが必要となる。節2-2でメニュー画面について説明し、節2-3および節2-4でプレイリスト間の結びつき（リンク）を定義するシナリオについて説明する。

## 2-2. メニュー画面について

グラフィクスプレーン12には、ユーザに対して操作を促す画面、例えばメニュー画面を表示させることができる。第25図は、グラフィクスプレーン12に表示されるメニュー画面60の一例を示す。メニュー画面60は、文字や画像を特定の位置に表示させ、ユーザが選択や実行を指示することで新たな動作が開始される「リンク」や「ボタン」を配置することができる。

「リンク」は、例えば、文字列や画像データに対して所定のファイルへのアクセス方法を示す記述を設定し、ユーザがポインティングデバイスなどを用いて画面に表示された当該文字列や画像データ上を指定することで、当該文字列や画像データに設定されたアクセス方法に従って当該ファイルにアクセスできるようにしたものである。「ボタン」は、例えば、「リンク」において通常の状態、選択された状態および押された状態の3種類の画像データを用意し、ユーザにより当該ボタン画像が指定された際に、3種類の画像データの表示を切り換え



て、ユーザが行った操作をユーザ自身に認識しやすくしたものである。

「リンク」や「ボタン」に対する指定動作は、マウスを用いて画面  
上のカーソル表示を移動させ、「リンク」による文字列や画像、「ボ  
タン」による画像上で、マウスボタンを1または複数回押すクリック  
5 動作を行うことでなされる。マウス以外のポインティングデバイスに  
よっても、同様の動作を行うことができる。これに限らず、リモート  
コントロールコマンドやキーボードのキー操作により、「リンク」や  
「ボタン」の指定を行うこともできる。方向キーなど所定のキーで指  
10 定を行いたい「リンク」や「ボタン」を選択し、決定キーなどにより  
選択された「リンク」や「ボタン」を指定する。

第25図の例では、グラフィクスプレーン12上に表示されたメニ  
ュー画面60の上部に、画像データとしてタイトル61が表示される  
。タイトル61に続けて、リンクとしての選択項目62A、62B、  
15 62Cおよび62Dが表示される。例えばリモートコントロールコマ  
ンドのキー操作により、選択項目62A、62B、62Cおよび62  
Dのうち1つを選択し指定することで、指定された選択項目にリンク  
として設定されたファイルがアクセスされる。

メニュー画面60の下部に、字幕の表示の有無や、出力される音声  
20 を例えば英語および日本語から選択するためのボタン64および65  
が表示される。これらボタン64や65を上述のように操作すること  
で、それぞれの設定を行う画面を表示するためのファイルがアクセス  
され、所定の画面表示がなされる。

メニュー画面60の左下部分には、項目の選択方法を示す文字列6  
25 3が表示される。この文字列63も、グラフィクスプレーン12上の  
表示である。

第25図に一例が示されるようなメニュー画面60を表示するためには、画面の表示方法やリンク情報などを記述可能な、何らかの記述言語が必要とされる。Blu-ray Discによる、このようなメニュー画面表示を可能とする記述言語としては、様々な方式のものが考えられるが、以下の説明では、下記の2種類の記述言語を用いた例について、それぞれ説明する。

- (1) DVDビデオのナビゲーションコマンドを基に変更・拡張を加えたコマンド体系に、字幕・ボタン用の独自の表示制御コマンドを加えた命令群を追加した記述言語。以下、独自コマンドと称する。
- (2) インターネット上のWWW(World Wide Web)などで広く普及している記述言語である、HTML(Hyper Text Markup Language)と、HTMLと親和性の高いスクリプト言語であるECMASクリプト。

### 2-3. シナリオについて

- さて、前節で述べたようなBlu-ray Discのメニュー画面60では、例えばプレイリストの一覧が画像データや文字列、ボタンなどで表示され、あるプレイリストを指定することで、指定されたプレイリストがディスクから読み出され再生されることが期待される。
- 第25図の例では、メニュー画面60に対してプレイリストの一覧が表示される。このメニュー画面60およびメニュー画面60上の選択項目を選択した際に再生される映像・音声は、実際には、複数のプレイリストが並んだ構造によって構成されている。メニューの一項目を構成する複数のプレイリスト内で、プレイリストの関連付けを行うことで、ストーリーを分岐させる仕組みを実現することができる。ストーリーを分岐できるようにすることで、ユーザの選択によってスト

ーリーが変化していくマルチストーリー機能や、プレーヤの設定言語に従って自動的に適切な言語を再生する機能や、ユーザの年齢に応じてシーンを差し替えるパレンタル機能を実現することができる。

このような機能は、記録済みのディスクにおいて特に有用であって  
5、テレビジョン放送の記録／再生が主目的である現行のBlu-ray Disc規格では想定されていない。

以下では、この複数のプレイリストが並べられた構造を、シナリオと称する。第26図は、独自コマンドを用いた場合のシナリオ70の一例の内部構造を示す。このシナリオ70は、複数のプレイリスト7  
10 3A～73Mを含む。また、シナリオ70は、グラフィクスプレーン12を用いて分岐選択画面を表示する箇所が2箇所（画面80Aおよび80B）設けられている。画面80Aは、分岐選択画面を表示するためのグラフィクスデータ74Aとプレイリスト73Cを有する。同様に、画面80Bは、分岐選択画面を表示するためのグラフィクスデータ74Bとプレイリスト73Jを有する。  
15

シナリオは、プレイリストの並びを決めると共に、グラフィクスプレーン12に対する表示のタイミングを規定する。グラフィクスプレーン12への表示のタイミングは、グラフィクスプレーンに表示する画像に対して付加された表示制御命令によって規定することができる  
20。

第26図の例では、メニュー画面60はシナリオ70における画面80Aに対応する。メニュー画面60における選択項目（例えば選択項目62A）は、グラフィクス74Aによって構成されている。メニュー画面60において選択項目62Aが指定されると、選択項目に対応したプレイリスト73Dが再生されることになる。  
25

第26図のシナリオ70では、まず、ディスクがプレーヤに挿入さ

れると、プレイリスト 7 3 A が再生される。プレイリスト 7 3 A の再生が終了されると、続けてプレイリスト 7 3 B が再生され、プレイリスト 7 3 B の再生が終了されると、プレイリスト 7 3 C が再生され、グラフィクスデータ 7 4 A が読み出されて、ユーザにストーリーの分岐選択を促す画面 8 0 A が表示される。

画面 8 0 A が表示された以降は、ユーザの選択に基づきストーリーが分岐される。この第 2 6 図の例では、第 1 の選択により、画面 8 0 A の後、プレイリスト 7 3 D、7 3 E、7 3 F の順に再生され、全体の再生が終了される。プレイリスト 7 3 F の再生が終了された後、メインのメニュー画面（例えば上述したメニュー画面 6 0）に戻るようにしてもよい。

画面 8 0 A における第 2 の選択では、画面 8 0 A の後、プレイリスト 7 3 G が再生される。プレイリスト 7 3 G は、所定のタイミングにマークが設定されていてもよい。プレイリスト 7 3 G が再生されると、再生装置自身の設定やユーザによる他のシナリオや分岐選択画面などでの選択に基づき、プレイリスト 7 3 G のマーク位置で分岐するか、プレイリスト 7 3 G を最後まで再生するかが制御される。プレイリスト 7 3 G が最後まで再生される場合には、プレイリスト 7 3 G の再生終了後、プレイリスト 7 3 M、7 3 I と再生され、プレイリスト 7 3 J がさらに再生される。

プレイリスト 7 3 G においてマーク位置で分岐された場合には、次にプレイリスト 7 3 K、7 3 L と再生され、プレイリスト 7 3 L の再生が終了されたら、次に、プレイリスト 7 3 I に設定されたマーク位置から再生が継続される。

プレイリスト 7 3 J は、グラフィクスデータ 7 4 B が読み出されて、ユーザにストーリーの分岐選択を促す画面 8 0 B が表される。画面

80Bにおける第1の選択では、プレイリスト73Fが再生される。  
画面80Bの第2の選択では、プレイリスト73Kに設定されている  
マーク位置からの再生が行われる。

- シナリオの再生において、マークの検出、ユーザからの入力、プレーヤの動作変更の検出がされた時の動作は、プレイリスト毎にコマンド列（プログラム）が用意されており、そのプログラムをプレーヤが実行することにより行われる。

- 次に、記述言語としてHTMLおよびECMASクリプトを用いた場合のシナリオ構造について、第27図を用いて説明する。第27図  
10 において、第26図と共通する部分には同一の符号を付し、繁雑さを避けるために詳細な説明を省略する。

- 第27図に示されるシナリオ70'は、第26図におけるシナリオ70と同様のメニューや分岐選択を有し、シナリオ70と同様に進行可能である。シナリオ70'は、複数のプレイリスト73A~73M  
15 を含み、グラフィクスプレーン12を用いて分岐選択画面を表示する箇所が2箇所（画面80Aおよび80B）設けられる。

- 画面80Aは、分岐選択画面を表示するためのグラフィクスデータ74Aとプレイリスト73Cを有する。画面80Bは、分岐選択画面を表示するためのグラフィクスデータ74Bとプレイリスト73Jを  
20 有する。シナリオ70'は、プレイリストの並びを決めると共に、グラフィクスプレーン12に対する表示のタイミングを規定する。グラフィクスプレーン12への表示タイミングは、プレイリスト中のマークを用いて規定することができる。

- マークの検出、ユーザからの入力、プレーヤの動作変更の検出は、  
25 イベントドリブンモデルに基づきなされる。プレイリストの再生が開始された、プレイリストの再生が終了された、プレイリストの再生中

にマークが検出された、リモートコントロールコマンドのキー操作などによりユーザからの入力があった、などによりイベントが発生される。特定のイベントがきっかけになって実行されるイベントハンドラをプログラム中に用意しておくことで、プレーヤにおいて、当該イベントにより期待される動作が実行される。

第27図に一例が示されるシナリオ70'には、2つのイベントハンドラ71および72が設定されている。これらのうち、イベントハンドラ71は、シナリオ70の全体にわたって有効なイベントハンドラが記述される、グローバルなイベントハンドラ定義である。

例えば、シナリオ70'内のプレイリスト73A~73Mの何れが再生中であっても、リモートコントロールコマンドに設けられたメニューボタンが押されたら、シナリオの一覧が表示されるメニュー画面60が表示される。この、メニュー画面60を表示するためのプレイリストに再生処理が移行される動作を実現したい場合について考える。この場合、リモートコントロールコマンドのメニューボタンが押されたときに発生するイベント（メニューボタン押下イベント）に対応するイベントハンドラであり、メニュー画面60を表示させるプレイリストに処理が移行される命令を、グローバルなイベントハンドラ71として記述する。

イベントハンドラ72は、特定のプレイリストの再生中にのみ、あるいは、特定のユーザ入力画面が表示されている間のみに実行され得るイベントハンドラが記述される、ローカルなイベントハンドラ定義である。例えば、分岐選択画面である画面80Aに表示されているリンクがユーザにより指定されたときに、別のプレイリストを再生するという動作は、リンクが指定されたというイベントに対して、プレイリストの再生を開始する命令をローカルなイベントハンドラ72とし

て記述するすることで、実現することができる。

このようなイベントハンドラ定義は、E C M Aスクリプトを用いて記述される。E C M Aスクリプトは、E C M A (European Computer Manufacturers Association)により定められた、J a v a S c r i p t (登録商標) に基づいたクロスプラットフォーム用のスクリプト言語である。E C M Aスクリプトは、HTML文書との親和性が高いことと、独自のオブジェクトの定義が可能である。

#### 2-4. シナリオの分類

- 10 シナリオは、後述するように、B D M Vディレクトリに一つ定義され、1または複数のプレイリストから構成される。ここで、シナリオの構成の分類について第28図および第29図A、第29図Bおよび第29図Cを用いて説明する。シナリオの構成は、プレイリスト同士の接続を基に考えると、第28図に示されるように、(1) シングル  
15 プレイリスト、(2) シーケンシャルプレイリスト、(3) マルチプレイリスト、の3種類に大別することができる。

- (1) のシングルプレイリストは、第29図Aに一例が示されるように、単一のプレイリストで構成されるシナリオである。タイムラインが定義可能で、シナリオ再生中の割り込みが無い。コンテンツが映画  
20 画であれば、ディスクのローディング後に映画本編だけが再生されるようなシナリオである。

- (2) のシーケンシャルプレイリストは、第29図Bに一例が示されるように、複数のプレイリストが分岐無しに直列的に並べられて構成されるシナリオである。各プレイリストは、プレイリストの末尾と  
25 次のプレイリストの先頭とが接続されるように並べられる。各プレイリストを通じてのタイムラインが定義可能である。コンテンツが映画

であれば、メニュー画面と映画本編との組み合わせからなるようなシナリオである。ディスクのローディング後にメニュー画面を表示させるプレイリストが実行され、メニュー画面に対して映画本編の再生が指示されると、次のプレイリストが実行され、映画の本編が再生される。

(3) のマルチプレイリストは、第29図Cに一例が示されるように、プレイリストの分岐や収束による接続を含むシナリオである。プレイリストを通じてのタイムラインを定義することができず、例えば各プレイリスト内にタイムラインが定義される。マルチプレイリストによれば、ユーザ入力によって再生内容が変化するようなインタラクティブ性やゲーム性を実現することができる。コンテンツが映画の場合には、同一シーンを様々なアングルで収録し、再生時にユーザの指示により選択して鑑賞できるようにしたマルチアングルを実現するようなシナリオである。

再生専用メディアの場合、シナリオは、BDMVディレクトリに一つ定義されるが、ユーザに対しては、シナリオをより細分化した単位を見せる必要がある。但し、プレイリストの単位がユーザの認識する単位と一致するとは限らない。例えば、一つのプレイリストに3本の映画が収録されているときには、各映画の頭出し点をユーザに見せる必要がある。プレイリストの構造とは独立した頭出し点（エントリポイント）を、タイトルおよび／またはチャプタと呼ぶことにする。

第30図を用いてタイトルおよびチャプタについて説明する。タイトルは、シナリオ中の任意の再生開始点を表す。この第30図の例では、プレイリスト470Aの先頭にタイトル1に設定されている。タイトル2は、プレイリスト470Dの途中に設定されている。プレイリスト470Aの先頭からタイトル2の直前までがタイトル1とされ



る。チャプタは、タイトルをさらに細分化した単位で、これもユーザが再生開始点として認識できる単位となる。タイトル1がチャプタによりさらに細分化される。第30図の例では、タイトル1に対してチャプタ1、2および3が設定され、タイトル1が3つに細分化されている。また、第30図に示されるように、タイトルおよびチャプタの位置は、プレイリストの途中を指定することもできる。

## 2-5. 仮想プレーヤモデルについて

次に、シナリオ記述に従って動作する再生装置のモデルを考える。

10 このモデル化された再生装置をBD (Blu-ray disc) 仮想プレーヤと称し、このBD仮想プレーヤの構成の定義をBD仮想プレーヤモデルと称する。

第31図を用いて、BD仮想プレーヤモデルについて説明する。BD仮想プレーヤ30は、ディスクのローディング後、この発明で定義したシナリオ記述言語により記述されディスクに記録されたシナリオを、PBCプログラム40として読み込み、シナリオの記述に従い動作する。

BD仮想プレーヤ30は、この発明の実施の一形態により定義されるディスク状記録媒体を再生するもので、パーソナルコンピュータなどのコンピュータ環境上のオブジェクトである。コンピュータ環境は、汎用的なパーソナルコンピュータに限られず、この発明の実施の一形態により定義されるディスク状記録媒体などを再生するように専用の構成された再生装置および／または記録再生装置に組み込まれたソフトウェア環境も含む。以下、この発明の実施の一形態により定義されるディスク状記録媒体を、ディスクと略称する。

BD仮想プレーヤ30の状態は、概略的には、プレイリストおよび

グラフィックスを再生する状態Aと、再生が停止された状態Bの2つの状態に分類できる。ある状態から別の状態への遷移や、ある状態の中での次の動作の指定は、BD仮想プレーヤ30のオブジェクトに対するコマンドにより行われる。

- 5      状態Aは、複数の動作を含む。状態Aに含まれる動作としては、高速再生、逆転再生といった変速再生や、ディスク上の任意の時刻から再生する飛び込み再生といった特殊再生などが考えられる。BD仮想プレーヤ30においてグラフィックスプレーン12による表示がなされているときは、これら変速再生や特殊再生が制限されることがある。
- 10      PBC (Play Back Control) プログラム40は、例えばディスクに記録されているシナリオに対応する。詳細は後述するが、シナリオは、ディスクに記録されたプレイリストの再生方法や、メニュー画面の表示を指示する。PBCプログラム40とBD仮想プレーヤ30とは、API (Application Programming Interface) 41を介してコマン
- 15      ドのやりとりを行い、ディスクに記録されたプレイリストを再生する。

- より具体的には、BD仮想プレーヤ30の状態遷移の際には、PBCプログラム40により、API 41を介して、BD仮想プレーヤ30において使用目的が定められたメモリとして定義された共通パラメータ32に対して、必要な情報が伝達される。この共通パラメータ32
- 20      に対する値のセットは、PBCプログラム40とBD仮想プレーヤ30との間でAPI 41を介してやりとりされるコマンドによって直接、あるいは、API 41を介して実行されるプレーヤコマンド31から間接的に行うことができる。

- 25      この発明の実施の一形態では、BD仮想プレーヤ30は、イベントドリブンモデルに基づき制御される。BD仮想プレーヤ30の動作中

には、様々なイベントが発生する。イベントは、ユーザによるキー入力やリモートコントロールコマンドによる操作、タイマによる割り込みなどにより、ハードウェア／OS (Operating System) 50において発生され、BD仮想プレーヤ30に渡される。これに限らず、再生されたプレイリスト中のマークの検出や、プレーヤの動作変更の検出など、BD仮想プレーヤ30自身によりイベントが発生される場合もある。

発生されるイベントの種類は、BD仮想プレーヤモデルで定義されている。イベントが発生すると、発生したイベントに対応するイベントハンドラが実行される。BD仮想プレーヤに組み込まれているイベントハンドラが実行され、規格で規定されているプレーヤ動作が実行される。

BD仮想プレーヤ30における割り込みイベントは、(1)再生中のコンテンツから発せられるイベント、(2)ユーザからの割り込みにより発生されるイベント、(3)プレーヤの状態の変化により発生されるイベント、の3種類に大別される。

(1)の、再生中のコンテンツから発せられるイベントは、予め設定された割り込みであって、当該コンテンツの再生の度に、同一のタイミングで発生される。例えば、BD仮想プレーヤ30においてプレイリストを再生中に、プレイリスト中に記述されたマークで指し示したディスク上の時刻に到達すると、BD仮想プレーヤ30においてマーク検出の割り込みが発生される。スクリプトにおいてタイマが設定された場合に、設定された時刻や、当該スクリプトによりタイマ設定がなされた10秒後などにタイマ割り込みのイベントが発生される。

(2)の、ユーザからの割り込みによるイベントは、発生および発生タイミングの予測ができないイベントである。例えば、ユーザによ

るリモートコントロールコマンドのキー操作により、この割り込みイベントが発生される。この場合、ユーザが何時キー操作を行うかが分からないため、事前にそのタイミングを知ることができない。

(3) の、BD 仮想プレーヤ 30 の状態変化によるイベントは、例えば、音声や字幕のストリーム変更を通知するためのイベントである。コンテンツの再生状態から停止状態へ、停止状態から再生状態への状態遷移でも、このイベントが発生される。このプレーヤの状態変化によるイベントは、上述した(1)の再生中のコンテンツから発せられるイベントや、(2)のユーザによる割り込みイベントに伴い引き起こされる場合も多い。上述の音声や字幕のストリーム変更の通知のイベントの例では、ユーザによるリモートコントロールコマンドなどのキー操作による割り込みイベントで、音声や字幕のストリームが変更され、この変更に伴うBD 仮想プレーヤ 30 の状態遷移に基づき、変更通知のイベントが発生される。

記述言語としてHTML およびECMA スクリプトを用いた場合、グラフィクスプレーン 12 を用いた表示制御がHTML (Hyper Text Markup Language) 4.0 またはXHTML (eXtensible HTML) 文書として記述され、グラフィクスプレーン 12 による表示画面に対するイベントは、HTML 4.0 の組み込みイベントが用いられる。また、HTML 4.0 の組み込みイベントでは不十分なイベントは、ECMA スクリプトを用いて記述される。

HTML 形式とECMA スクリプトを組み合わせて用いる場合、イベントが発生すると、まず、そのイベントが発生した要素に属性で指定されているイベントハンドラが文書中に存在するか否かで、処理が異なる。イベントハンドラが存在する場合は、そのイベントハンドラが実行される。イベントハンドラが存在しない場合は、文書中にグロ

ーバルなイベントハンドラが用意されているか否かが探される。その結果、若し、グローバルなイベントハンドラが当該文書中に存在すれば、そのイベントハンドラが実行される。文書中にスクリプト言語によるイベントハンドラの記述が無い場合には、BD仮想プレーヤ30

5 において当該イベントに対して用意されたデフォルトのイベント処理が行われる。

イベントハンドラの記述方法としては、HTML文書における要素の属性として記述する方法と、ECMASクリプトにおいてメソッドcaptureEventsを使って記述する方法とを用いることができる。

10 HTML文書を用いてイベントハンドラを記述する方法について説明する。この発明の実施の第1の形態によるBD仮想プレーヤ30では、HTML4.0に規定される組み込みイベントのうち、例えばイベントonload、イベントonunload、イベントonclickおよびイベントonkeypressを用いることができる。これらのイベントは、タグの要素

15 中に属性として記述される。

イベントonloadは、利用者エージェントが1つのウィンドウまたは1対のタグ<FRAMESET></FRAMESET>で定義される全てのフレームを終了する際に発生される。例えば、メニュー画面の表示の際にイベントonloadが発生される。

20 ウィンドウは、HTMLの規定に基づきHTMLファイルを表示するブラウザアプリケーションにおける表示単位であり、フレームは、1つのウィンドウを分割して複数のHTMLファイルを表示させる際に用いられる枠組みである。フレーム内に表示されるHTMLファイルも含めてフレームと称する。このイベントonload属性は、要素BODY

25 および要素FRAMESETと共に使用することができる。

イベントonunloadは、利用者エージェントが1つのウィンドウまた

は1つのフレームから1つのHTML文書を除去する際に発生される。このイベントonunload属性は、要素BODYおよび要素FRAMESETと共に使用することができる。

5 イベントonclickは、ポインティングデバイスなどにより要素が指定された場合に発生される。例えば、マウスボタンのクリック操作を行った場合に、イベントonclickが発生される。このイベントonclick属性は、HTML 4.0における殆どの要素と共に用いることができる。

10 イベントonkeypressは、キーが要素上で押されたり解放されたりする際に発生される。例えば、ある要素により定義される画面上の領域が選択状態のときに、キーボード上の所定のキーやリモートコントロールコマンドのキーが押された際に、イベントonkeypressが発生される。このイベントonkeypress属性は、HTML 4.0における殆どの要素と共に用いることができる。

15 BD仮想プレーヤ30の動作を制御するためには、上述のHTMLによるイベントでは不十分であるため、独自のイベントを定義する必要がある。第32図A、第32図Bおよび第32図Cは、BD仮想プレーヤ30において独自に定義される一例のイベントを示す。このイベントは、ECMAScriptにより、HTML文書に埋め込まれて  
20 記述される。イベントハンドラを指定する属性名は、イベント名の先頭に「on」を付して示す。

イベントTimerFiredは、カウントダウンタイマの値が「0」になるか、カウントアップタイマの値が指定された値になったときに発生される。イベントPlayStoppedおよびイベントPlayStilledは、それぞれ  
25 再生の停止および一時停止により発生される。イベントStillReleasedは、一時停止が解除された際に発生される。イベントPlayPausedお

よびイベントPauseReleasedは、それぞれユーザによる再生の一時停止および一時停止の解除の際に発生される。イベントPlayStartedは、再生が開始された際に発生される。イベントPlayRepeatedは、繰り返し再生時の先頭が検出された際に発生される。

- 5 イベントSPDisplayStatusChangedは、サブピクチャ（字幕）ストリームの表示／非表示状態が変化した際に発生される。イベントSelectedAudioChangedおよびイベントVideoStoppedは、それぞれ再生するオーディオストリームおよびビデオストリームが変わった際に発生される。
- 10 イベントScenarioStartedおよびイベントScenarioEndedは、それぞれシナリオの先頭および終端が検出された際に発生される。イベントPlayListStartedおよびイベントPlayListEndedは、それぞれプレイリストの先頭および終端が検出された際に発生される。イベントPlayItemStartedおよびイベントPlayItemEndedは、それぞれプレイアイテム
- 15 の先頭および終端が検出された際に発生される。

イベントMarkEncounteredは、プレイリストの再生中にマークが検出された際に発生される。このイベントは、例えばグラフィクスプレーン12に画像データを表示する際に用いられる。共通パラメータ32に、検出されたマークの種類と番号とが格納される。

- 20 イベントButtonPressedは、画面上に配置されたボタンが押された際に発生される。例えば、グラフィクスプレーン12上に表示されたボタンを、キー操作やマウスのクリック操作などにより仮想的に押した場合に、イベントButtonPressedが発生される。

- イベントValidPeriodStartedは、有効期間が開始された際に発生さ
- 25 れる。これは、リンクの選択有効期間を設定する際に用いることができる。イベントValidPeriodEndedは、有効期間が終了した際に発生さ

れる。これは、リンクの強制実行のときに用いることができる。

イベントKeyPressedは、リモートコントロールコマンドのキーが押されたときに発生される。押されたキーの種類は、イベントハンドラ内の「switch」文などで識別される。

- 5     シナリオ記述言語として独自コマンドを用いた場合、この独自コマンドとしてシナリオを実行するに当たって必要なイベントを予め言語として定義することができる。そのため、独自コマンドを用いてシナリオ記述を行う場合には、汎用なECMASクリプトを用いる場合のように、シナリオの実行に特に適合したイベントをプログラム内で定義する必要は、無い。
- 10

## 2-6. コマンドについて

- BD仮想プレーヤ30は、一連のコマンドを有し、これらのコマンドにより、BD仮想プレーヤ30における動作および状態、BD仮想プレーヤ30で扱うビデオストリーム、オーディオストリームおよびサブピクチャ（字幕プレーン11の画像データ）に関する情報取得や制御、共通パラメータ32に対する操作、タイマやキー入力の割り込みに応じた処理、グラフィクスプレーン12で扱う画像データの制御がそれぞれ定義される。
- 15

- 20     これらのコマンドは、第31図で既に説明した、BD仮想プレーヤ30におけるAPI41に実装される。PBCプログラム40の記述に基づきこれらのコマンドがAPI41を介して呼び出され、BD仮想プレーヤ30におけるディスク再生が制御される。PBCプログラム40の具体的な例については、後述する。

- 25     BD仮想プレーヤ30が有するコマンドは、シナリオ記述言語として独自コマンドを用いる場合と、HTMLおよびECMASクリプト



を用いる場合とで、多少異なる。先ず、シナリオ記述言語として独自コマンドを用いる場合のコマンドについて、第 3 3 図 A、第 3 3 図 B、第 3 3 図 C、第 3 3 図 D、第 3 3 図 E、第 3 3 図 F、第 3 3 図 G および第 3 3 図 H を用いて説明する。

- 5 再生開始位置指定に関するコマンドについて説明する。コマンド `LinkPlayList(playListNumber)` で、"playListNumber" で指定されたプレイリストの再生開始が指示される。コマンド `LinkPlayItem(playListNumber,playItemNumber)` で、指定のプレイリストの指定のプレイアイテムからの再生開始が指示される。"playItemNumber" は、"PlayItem\_
- 10 id" であり、値が「0」から始まる。"playItemNumber" に値「0」を指定することで、当該プレイアイテムが属するプレイリストの先頭から再生される。

- コマンド `Link(position)(object)` で、シナリオ内での移動が指示される。現在移動中の箇所から前後のプレイリスト、プレイアイテムおよび
- 15 びチャプタに移動することがこのコマンドで指示される。なお、パラメータ "position" は、"prev"、"next"、"top"、"Parent" または "tail" の何れかの値を取り、パラメータ "object" で示される移動対象（プレイリスト、プレイアイテムまたはチャプタ）に対する移動方法が指示される。

- 20 コマンド `Exit` で、シナリオ再生の停止が指示される。この場合、標準レジスタの値は、保持されない。コマンド `RSM` で、プレーヤ内のメモリに保存されているレジューム情報を呼び出し、レジスタにセットして再生を開始する。

- プレーヤ状態の取得に関するコマンドについて説明する。コマンド
- 25 `getMenuDescriptionLanguage()` で、メニュー表示の際に用いられる言語が取得される。コマンド `getScenarioNumber()`、コマンド `getPlayLi`

stNumber() およびコマンド getChapterNumber() で、再生中のシナリオ番号、プレイリスト番号およびチャプタ番号がそれぞれ取得される。コマンド getPlayerSupport() で、プレーヤのバージョン情報が取得される。

- 5     ビデオストリームに関するコマンドについて説明する。コマンド getVideoStreamAvailability() で、指定のビデオストリームが含まれているか否かが取得される。コマンド setVideoStreamNumber() で、デコードするビデオストリームが指定される。コマンド getVideoStreamNumber() で、選択中のビデオストリームの番号が取得される。コマンド
- 10   getVideoStreamAttribute() で、選択中のビデオストリームの属性が取得される。ビデオストリームの属性は、例えば、そのビデオストリームの符号化方式、解像度、アスペクト比、アスペクト比が 4 : 3 のときのディスプレイモード、クローズドキャプションの有無である。コマンド setAngleNumber() で、アングル番号が指定される。コマンド
- 15   getAngleNumber() で、選択中のアングル番号が取得される。コマンド getMaxVideoStreams() で、ビデオストリームの最大数が取得される。

- オーディオストリームに関するコマンドについて説明する。コマンド getAudioStreamAvailability() で、指定のオーディオストリームが含まれているか否かが取得される。コマンド getAudioStreamLanguage
- 20   () で、指定のオーディオストリームの言語に関する情報が取得される。コマンド getAudioStreamStatus() で、指定のオーディオストリームの状態が取得され、コマンド setAudioStreamStatus() で、指定のオーディオストリームに対する状態のセットが指示される。オーディオストリームの状態は、例えば再生または非再生である。コマンド getAudioStreamAttribute() で、指定のオーディオストリームの属性が取得
- 25   される。

サブピクチャストリーム（字幕データ）ストリームに関するコマンドについて説明する。コマンドgetSPStreamAvailability()で、指定のサブピクチャストリームが含まれているか否かが取得される。コマンドgetSPStreamLanguage()で、指定のサブピクチャストリームで用いられている言語が取得される。コマンドgetSPDisplayStatus()で、サブピクチャストリームの表示状態が取得され、コマンドsetSPDisplayStatus()で、サブピクチャストリームの表示状態がセットされる。サブピクチャストリームの表示状態は、例えばサブピクチャストリームの表示のオン／オフである。コマンドgetSPStreamAttribute()で、指定のサブピクチャストリームの属性が取得される。サブピクチャストリームの属性は、例えばサブピクチャストリームの表示がアスペクト比4：3またはワイド画面の何れでなされるかである。

共通パラメータ32に関するコマンドについて説明する。図中では、レジスタリード／ライトとして記されている。コマンドclearReg()で、BD仮想プレーヤ30が内蔵しているメモリ領域すなわちレジスタについて、全レジスタの初期化が指示される。コマンドsetReg()で、指定のレジスタへの値のセットが指示される。コマンドgetReg()で、指定のレジスタからの値の読み出しが指示される。

タイマに関するコマンドについて説明する。コマンドsleep()で、指定された時間の処理の停止が指示される。コマンドsetTimeout()で、指定された時間後に関数や処理を実行することが指示される。コマンドsetInterval()で、指定された時間毎に処理を実行することが指示される。タイマに関するコマンドにおいて、時間はミリ秒単位で指定することができる。コマンドclearTimer()で、指定された登録タイマIDの処理の中止が指示される。コマンドpauseTimer()で、指定された登録タイマIDのタイマの一時停止が指示される。コマンドresu

meTimer()で、指定された登録タイマIDのタイマを一時停止から再開することが指示される。

効果音に関するコマンドでは、コマンドplaySoundEffect(sound\_id)で、指定の効果音を再生する。

- 5     この第33図A、第33図B、第33図C、第33図D、第33図E、第33図F、第33図Gおよび第33図Hに例示されるようなコマンドを、後述するポストコマンド領域やボタンコマンド領域に書くことにより、特定のプレイリストへのジャンプを実現することができる。コマンドは、この第33図A、第33図B、第33図C、第33  
10 図D、第33図E、第33図F、第33図Gおよび第33図Hに例示されるに止まらず、さらに他のコマンドを定義することも可能である。

- シナリオ記述言語としてHTMLおよびECMASクリプトを用いる場合のコマンドについて、第34図A、第34図B、第34図C、  
15 第34図D、第34図E、第34図F、第34図G、第34図H、第34図Iおよび第34図Jを用いて説明する。

- プレーヤ動作に関するコマンドについて説明する。コマンドplayScenario(scenarioNumber,[scenarioTime])で、"scenarioNumber"で指定されるシナリオの再生が指示される。"scenarioNumber"は、実際に  
20 は、シナリオ構成を記述したファイルなどの場所を示すURI(Universal Resource Identifier)になる。コマンドplayPlayList(playListNumber)で、"playListNumber"で指定されたプレイリストの再生が指示される。コマンドplayChapterMark(playListNumber,chapterNumber)で、"playListNumber"で示されるプレイリスト中の、"chapterNumber"  
25 r"指定されるチャプタからの再生が指示される。コマンドplayPlayItem(playListNumber,playItemNumber)で、"playListNumber"および"pl

ayItemNumber"で示されるプレイアイテムからの再生が指示される。"playItemNumber"は、"playItem\_id"であり、値「0」を指定することで、当該プレイアイテムが属するプレイリストの先頭から再生される。

- 5     コマンドplay(position)(object)で、シナリオ内での移動が指示される。現在移動中の箇所から前後のプレイリストやプレイアイテムに移動することがこのコマンドで指示される。パラメータ"position"は、"prev"、"next"、"top"、"goUp"または"tail"の何れかの値を取り、パラメータ"object"で示される移動対象（プレイリスト、プレイアイテムまたはチャプタ）に対する移動方法が指示される。

      コマンドstop()で、シナリオ再生の停止が指示される。この場合、標準レジスタの値は、保持されない。コマンドresume()で、前回再生を停止した箇所からの再生の再開が指示される。コマンドplaySoundEffect()で、選択された効果音の再生が指示される。

- 15     プレーヤ状態に関するコマンドについて説明する。コマンドgetMenuDescriptionLanguage()で、メニュー表示の際に用いられる言語が取得される。コマンドgetScenarioNumber()、コマンドgetPlayListNumber()およびコマンドgetChapterNumber()で、再生中のシナリオ番号、プレイリスト番号およびチャプタ番号がそれぞれ取得される。コマンドgetPlayerSupport()で、プレーヤのバージョン情報が取得される。

- ビデオストリームに関するコマンドについて説明する。コマンドsetVideoStreamNumber()で、デコードするビデオストリームが指定される。コマンドgetVideoStreamNumber()、コマンドgetVideoStreamStatus()およびコマンドgetVideoStreamAttr()で、再生中のビデオストリームの番号、状態および属性がそれぞれ取得される。なお、ビデオストリームの属性は、例えば、そのビデオストリームの符号化方式、解
- 25

像度、アスペクト比、アスペクト比が4 : 3のときのディスプレイモード、クローズドキャプションの有無である。コマンドsetAngleNumber()で、アングル番号が指定される。コマンドgetAngleNumber()で、選択中のアングル番号が取得される。コマンドgetMaxVideoStream()

5 で、ビデオストリームの最大数が取得される。

オーディオストリームに関するコマンドについて説明する。コマンドgetAudioStreamAvailability()で、指定のオーディオストリームが含まれているか否かが取得される。コマンドgetAudioStreamLanguage()  
10 。コマンドgetAudioStreamStatus()で、指定のオーディオストリームの状態が取得され、コマンドsetAudioStreamStatus()で、指定のオーディオストリームに対する状態のセットが指示される。オーディオストリームの状態は、例えば再生または非再生である。コマンドgetAudioStreamAttribute()で、指定のオーディオストリームの属性が取得  
15 される。

サブピクチャストリーム（字幕データ）ストリームに関するコマンドについて説明する。コマンドgetSPStreamAvailability()で、指定のサブピクチャストリームが含まれているか否かが取得される。コマンドgetSPStreamLanguage()で、指定のサブピクチャストリームで用  
20 いられている言語が取得される。コマンドgetSPDisplayStatus()で、サブピクチャストリームの表示状態が取得され、コマンドsetSPDisplayStatus()で、サブピクチャストリームの表示状態がセットされる。サブピクチャストリームの表示状態は、例えばサブピクチャストリームの表示のオン／オフである。コマンドgetSpStreamAttribute()で、  
25 指定のサブピクチャストリームの属性が取得される。サブピクチャストリームの属性は、例えばサブピクチャストリームの表示がアスペク

ト比 4 : 3 またはワイド画面の何れでなされるかである。

- 共通パラメータ 3 2 に関するコマンドについて説明する。第 3 4 図 A、第 3 4 図 B、第 3 4 図 C、第 3 4 図 D、第 3 4 図 E、第 3 4 図 F、第 3 4 図 G、第 3 4 図 H、第 3 4 図 I および第 3 4 図 J では、レジスタリード／ライトとして記されている。コマンド `clearReg()` で、B D 仮想プレーヤ 3 0 が内蔵しているメモリ領域すなわちレジスタについて、全レジスタの初期化が指示される。コマンド `setReg()` で、指定のレジスタへの値のセットが指示される。コマンド `getReg()` で、指定のレジスタからの値の読み出しが指示される。
- 10 タイマに関するコマンドについて説明する。コマンド `sleep()` で、指定された時間の処理の停止が指示される。コマンド `setTimeout()` で、指定された時間後に関数や処理を実行することが指示される。コマンド `setInterval()` で、指定された時間毎に処理を実行することが指示される。タイマに関するコマンドにおいて、時間はミリ秒単位で指定することができる。コマンド `clearTimer()` で、指定された登録タイマ ID の処理の中止が指示される。コマンド `pauseTimer()` で、指定された登録タイマ ID のタイマの一時停止が指示される。コマンド `resumeTimer()` で、指定された登録タイマ ID のタイマを一時停止から再開することが指示される。
- 20 キー入力に関するコマンドでは、コマンド `getPressedKey()` で、入力された（押された）キーの種類が取得される。

- グラフィクスに関するコマンドについて説明する。コマンド `loadGraphics(htmlfile, ID)` で、"htmlfile" で示されるファイルを読み込み、グラフィクスプレーン 1 2 に非表示で展開することが指示される。
- 25 展開されたグラフィクス画像には "ID" が割り当てられ、後述のコマンドで参照される。コマンド `showGraphics(ID)` は、上述のコマンド `load`

Graphics(htmlfile, ID)により既にグラフィクスプレーン1 2上に展開されている画像に対して、表示が指示される。コマンドhideGraphics(ID)で、"ID"で示される画像の表示を消去することが指示される。

その他のコマンドについて説明する。コマンドrandom(input Number num)で、1から"num"までの乱数の発生が指示される。乱数の発生は、独自の定義で行われる。コマンドcatchEvent(eventname, eventhandler)で、"eventname"で示されたイベントが発生した際、"eventhandler"で示された関数を実行することが指示される。

## 10 2-7. コマンドの実行について

上述のように定義されたコマンドの実行について説明する。まず、シナリオ記述言語として独自コマンドを用いた場合の例について説明する。シナリオ記述言語として独自コマンドを用いた場合、シナリオには、プレーヤの動作を指示するコマンドが並んだプログラムを含むコマンドを配置する領域として、グローバルコマンド領域と、ローカルコマンド領域の二つがある。

グローバルコマンド領域は、シナリオ全体に有効なプログラムが格納されており、例えば、ディスクがプレーヤに挿入されたときに最初に自動的にプレーヤによって実行されて、パラメータの初期化を行い、メニュー画面を構成するプレイリストにジャンプするようなプログラムを記述する領域である。ローカルコマンド領域は、各プレイリストに関係するプログラムを記述する領域である。ローカルコマンドは、さらにプレコマンド、プレイアイテムコマンド、ポストコマンド、ボタンコマンドの4種類に分類される。

25 第35図Aおよび第35図Bは、独自コマンドを記述言語として用いたシナリオで記述されるコマンドによるBD仮想プレーヤ30の動



作を概略的に示す。第35図Aは、ディスクのローディング時の動作の例を示す。既に述べたように、シナリオは、後述するBDMVディレクトリに対して一つが作られる。ディスクがプレーヤに挿入されディスクに対するイニシャルアクセスがなされると（ステップS30）

5、レジスタすなわち共通パラメータ32が初期化される（ステップS31）。そして、次のステップS32で、プログラムがディスクから読み込まれて実行される。イニシャルアクセスとは、ディスク挿入時のように、ディスクの再生が初めて行われることをいう。

この、ディスクローディング時に最初に読み込まれ、実行されるコマンド群（プログラム）を、グローバルコマンドと称する。グローバルコマンドは、例えば宣伝用映像（トレーラー）やメニュー画面を構成するプレイリストへのジャンプ命令が記述されており、その命令通りにプレーヤがプレイリストを再生していくことになる。

10

第35図Bは、プレーヤ30が停止状態からユーザにより例えばプレイキーが押下され再生が指示された場合の動作の例を示す。これは、第31図で説明したBD仮想プレーヤ30における状態Bから状態Aへの状態遷移に相当する。最初の停止状態（ステップS40）に対して、ユーザにより、例えばリモコンなどを用いて再生が指示される（UOP：User Operation）。再生が指示されると、先ず、レジスタ

15

20 すなわち共通パラメータ32が初期化され（ステップS41）、次のステップS42で、プレイリストの再生フェイズに移行する。

プレイリストの再生フェイズにおけるプレイリストの再生について、第36図Aおよび第36図Bを用いて説明する。第36図Aは、プレイリストが単一のプレイアイテムからなる場合の例である。プレイリストは、プレコマンド領域、プレイアイテムコマンド領域およびポストコマンド領域の3箇所にプログラムが配される。プレイリストの

25

再生フェイズに移行すると、先ずプレコマンド領域のプレコマンドが  
実行され（ステップS 1 0）、プレコマンドの実行が終了されると、  
プレイリストを構成するプレイアイテムの再生フェイズに移行する（  
ステップS 1 1）。プレイアイテムの再生フェイズでは、プレイアイ  
5 テムにおいて開始点および終了点で指定されたストリームが再生され  
る（ステップS 1 1 0）。終了点までの再生が終了した時点で、プレ  
イアイテムコマンドが実行される（ステップS 1 1 1）。プレイアイ  
テムコマンドが実行されると、次にポストコマンド領域のポストコマ  
ンドが実行され（ステップS 1 2）、プレイリストの再生が終了され  
10 る。

ポストコマンドは、一般的な使用においては、次に再生すべきプレ  
イリストへのジャンプ命令や、メニュー画面を構成するプレイリスト  
へのジャンプ命令が記述される。ジャンプ命令が無い場合には、そこ  
で再生が停止し、プレーヤは、停止状態（第31図の状態B）に遷移  
15 される。

第36図Bは、プレイリストが複数のプレイアイテムを含む場合の  
例である。この場合でも、プレイリスト中にプレコマンド領域、プレ  
イアイテムコマンド領域およびポストコマンド領域が配され、それぞ  
れプログラムが格納される。複数のプレイアイテムを含む場合、プレ  
20 イアイテムコマンド領域では、プレイアイテム数分のプレイアイテム  
ストリームおよびプレイアイテムコマンドが、時系列に従って配置さ  
れる。

プレイリストが複数のプレイアイテムを含む場合でも、プレイリス  
トの再生フェイズに移行すると、先ず、プレコマンドが実行される（  
25 ステップS 1 0）。次のプレイアイテムの再生フェイズでは、プレイ  
アイテムの開始点から終了点までのストリームの再生と、プレイアイ

テムコマンドの実行とが、プレイリストが含むプレイアイテムの数だけ実行される。第36図Bの例では、最初のプレイアイテムストリームが再生され（ステップS110-1）、対応するプレイアイテムコマンドが実行される（ステップS111-1）。次に、図示は省略するが、2番目のプレイアイテムストリームが再生され（ステップS110-2）、対応するプレイアイテムコマンドが実行される（ステップS111-2）。これをプレイアイテムの数だけ行い、最後のプレイアイテムストリームが再生され（ステップS110-n）、対応するプレイアイテムコマンドが実行されると（ステップS111-n）、プレイアイテムの再生フェイズが終了する。プレイアイテムの再生フェイズが終了すると、ポストコマンドが実行され（ステップS12）、このプレイリストの再生フェイズが終了される。

この発明の実施の一形態では、BD仮想プレーヤ30、ならびに、BD仮想プレーヤ30上で実行されるシナリオ、プレイリストおよびプレイアイテムを、階層的に考えることができる。すなわち、第37図Aで示されるように、BD仮想プレーヤ層600の下に1つのシナリオ層601があり、シナリオ層601の下に、1または複数のプレイリストを含むプレイリスト層602がある。プレイリスト層602の下には、プレイアイテム（PI）層603があり、プレイリスト毎に1または複数のプレイアイテムを持つことができる。

このような階層的な構造とすることで、プレイリストおよびプレイアイテムは、シナリオ層601を介してBD仮想プレーヤにより実行されることになる。したがって、シナリオにプレイリストに対する制御コマンドを記述することで、プレイリストの分岐などの制御が容易に実現される。これは、プレイアイテムに関しても同様である。この様子を第37図Bに示す。

シナリオ記述言語としてHTMLおよびECMASクリプトを用いた場合の例について説明する。以下の説明は、シナリオ記述言語としてHTMLおよびECMASクリプトを用いた場合の、より具体的なPBCプログラム40の例である。

- 5     HTMLおよびECMASクリプトを用いてシナリオを記述した場合、一つのシナリオに対して、一つのスクリプトファイルが作成される。グラフィクスプレーン12を用いてメニュー画面60などを表示する場合には、1画面に対して一つのHTMLファイルが作成される。スクリプトファイルおよびHTMLファイルは、ファイル名の拡張
- 10    子がそれぞれ「js」および「html」とされ、拡張子で以てこれら2種類のファイルが識別可能される。ディスクがドライブ装置に挿入されて最初に実行されるスクリプトプログラムが格納されるファイルは、ファイル名を例えば「startup.js」に固定的とする。

- 一例として、第38図に示されるようなシナリオ構成のディスクを
- 15    考える。このディスクは、シナリオScenario0000およびScenario0001の2つのシナリオを有し、シナリオScenario0000は、シナリオScenario0001へのリンクボタン92を表示するメニュー画面91を表示させる。ディスクが再生装置に装填されると、メニュー画面91が表示され、メニュー画面91に
- 20    表示されるボタン92が例えばクリック操作により押されると、シナリオScenario0001が再生される。シナリオScenario0001の再生が終了されると、再びメニュー画面91が表示される。

- 第39図は、第38図の構成に対して必要とされる一例のファイルを
- 25    一覧して示す。この例では、ファイル「startup.js」、ファイル「scenario000.js」、ファイル「000.html」、ファイル「00000.rpls」

、ファイル「scenario001.js」およびファイル「00001.rpls」の6ファイルが必要とされる。

これらのうち、ファイル「scenario000.js」は、シナリオScenario 000の構成情報が記述されるスクリプトファイルであって、メニュー画面91すなわちシナリオ一覧画面の構成情報が記述される。ファイル「000.html」は、メニュー画面91のレイアウト情報が記述されるHTMLファイルである。ファイル「00000.rpls」は、メニュー画面91において背景として表示されるプレイリストファイルである。ファイル「scenario001.js」は、シナリオScenario 001の構成情報が記述されるスクリプトファイルである。ファイル「00001.rpls」は、シナリオScenario 001で再生されるプレイリストの情報が記述されたプレイリストファイルである。

第39図では、プレイリストファイル「00000.rpls」および「00001.rpls」で実際に再生されるコンテンツファイルなど（クリップ情報ファイル「%%%%.clip」、AVストリームファイル「\*\*\*\*\*.m2ts」）については省略されている。

第39図に示される各ファイルは、第40図に一例が示されるディレクトリ構造に従い、ディスクに記録される。ディレクトリBD AVの直下にファイル「startup.js」が置かれると共に、ディレクトリBD AVの配下にディレクトリSCRIPTが設けられる。このディレクトリSCRIPTの下に、シナリオの構成情報が記述されるスクリプトファイル（ファイル「scenario000.js」など）や、メニュー画面のレイアウト情報などが記述されるHTMLファイル（ファイル「000.html」など）が置かれる。

第41図～第44図は、スクリプトファイルおよびHTMLファイルのより具体的な記述の例を示す。第41図は、ファイル「startup.

js」の記述例である。ファイル「startup.js」では、当該ディスクに格納されているシナリオの数と名前とが定義される。そして、メソッドplayScenario("scenario000")の記述により、ファイル「scenario000.js」が呼び出され、シナリオScenario000の再生が指示される。シナリオScenario000が再生されることにより、メニュー画面91が表示される。

第42図は、ファイル「scenario000.js」の記述例である。関数functionが定義され、メソッドcatchEvent()に「on」が付されて記述されたイベント（第32図A、第32図Bおよび第32図C参照）が発生すると、対応する関数functionで定義された内容が実行される。第42図の例では、メニュー画面91のレイアウト情報が記述されたHTMLファイル「000.html」が指定され、グラフィクスプレーン12へのメニュー画面91の読み込みおよび表示のタイミングが制御される。それと共に、メソッドplayPlayList("00000.rpls")が記述され、プレイリスト「00000.rpls」の再生が指示される。

このファイル「scenario000.js」により、プレイリスト「00000.rpls」による動画の動画プレーン10に対する表示が指示されると共に、プレイリスト「00000.rpls」の再生中に検出されたマークのタイミングで、メニュー画面91をグラフィクスプレーン12に表示する旨が指示される。

第43図は、ファイル「000.html」の記述例である。タグ<style type="text/css">および</style>で囲まれた部分に、「menu」および「scenario001」で参照される画像の、メニュー画面91におけるレイアウト情報が記述される。この第43図の例では、レイアウト情報が画面上の絶対座標で記述され、画像名「menu」で参照される画像データが画面の上端から200ピクセル、左端から

800ピクセルの位置に、幅200ピクセル、高さ50ピクセルの画像として表示されることが示される。同様に、画像名「scenario」で参照される画像データが画面の上端から700ピクセル、左端から700ピクセルの位置に、幅400ピクセル、高さ100ピクセルの画像として表示されることが示される。

タグ<script type="text/javascript">および</script>で囲まれた部分に、関数functionで、マウス操作に対するイベントハンドラonMoverhandler(f)、onMouthandler(f)およびonMclicckhandler(f)がそれぞれ定義される。この第43図の例では、イベントハンドラonMoverhandler(f)、onMouthandler(f)およびonMclicckhandler(f)に対してボタン画像を表示する画像データである画像データ「201.png」、「200.png」および「202.png」がそれぞれ対応付けられると共に、イベントハンドラonMclicckhandler(f)においては、シナリオファイル「scenario001.js」の再生が指示される。

タグ<body>および</body>で囲まれた部分に、メニュー画面91においてグラフィクスプレーン12に表示される画像データが指定される。上述のタグ<style type="text/css">および</style>で囲まれた部分に記述される画像名に対応する画像データのファイル名（「100.png」、「200.png」）がそれぞれ記述される。画像名「scenario001」で参照される画像データに対しては、マウスなどポインティングデバイスに対する操作に応じて発生するイベントonMouseover、onMouseoutおよびonclickに対して、上述の関数functionで定義されたイベントハンドラonMoverhandler(f)、onMouthandler(f)およびonMclicckhandler(f)がそれぞれ実行されることが指示される。

イベントonMouseoverは、指定された領域などにカーソル表示が重なった際に発生されるイベントである。イベントonMouseoutは、指定

された領域などからカーソル表示が離れた際に発生されるイベントである。イベントonclickは、指定された領域にカーソル表示があるときにポインティングデバイスに対する所定の操作、例えばマウスボタンのクリックが行われた際に発生されるイベントである。

- 5     第44図は、ファイル「scenario001.js」の記述例である。関数function UOPControl()において、このシナリオファイル「scenario001.js」の再生中に、例えばリモートコントロールコマンドに設けられたメニューキーが押されたときに、メニュー画面91を表示するためのシナリオScenario000を再生することが定義される。また、関数function playListEnded()により、このシナリオファイル「scenario001.js」によるプレイリストの再生が終了したら、メニュー画面91を表示するためのシナリオScenario000を再生することが定義される。このシナリオファイル「scenario001.js」でプレイリスト「00001.rpls」が再生されることが指示される。
- 10    第41図～第44図の記述に基づく動作について説明する。ディスクが再生装置に装填されると、ファイル「startup.js」が読み込まれ、このファイル中の記述によりファイル「scenario000.js」が呼び出される。このファイルに記述されたシナリオ「scenario000」が実行されると、第42図の記述に基づき、プレイリスト「00000.rpls」による動画が動画プレーン10に表示される。
- 15    プレイリスト「00000.rpls」中に指定されたマークに対応するタイミングで、ファイル「000.html」が呼び出され、ファイル「000.html」の記述に基づきシナリオ一覧を表示するメニュー画面91がグラフィクスプレーン12上に展開され、表示される。このメニュー画面91自体も、シナリオ「scenario000」という一つのシナリオで構成されたものである。
- 20



メニュー画面 9 1 には、例えば「Menu」という文字列が描画された画像ファイル「100.png」と、「Scenario 001」という文字列が描画された画像ファイル「200.png」とが配置されている。これらの画像ファイルは、グラフィクスプレーン 1 2 上に配置され表示される。グラフィクスプレーン 1 2 の背景に表示される動画プレーン 1 0 上では、プレイリスト「00000.rpls」による動画が表示される。動画プレーン 1 0 上のプレイリスト「00000.rpls」による動画と、グラフィクスプレーン 1 2 上のファイル「000.html」によるメニュー画面 9 1 とが重ね合わされて同一画面上に表示される。動画を背景にメニュー画面 9 1 が表示される。

このとき、グラフィクスプレーン 1 2 上に表示されている画面（メニュー画面 9 1）に所定に透明度を設定し、メニュー画面 9 1 を動画プレーン 1 0 上の動画に対して透過させて表示させることができる。この例では、プレイリスト「00000.rpls」の先頭および末尾にマークが設定され、メニュー画面 9 1 は、プレイリスト「00000.rpls」の再生と共に表示が開始され、プレイリスト「00000.rpls」の再生終了と共に表示が消去されるようになっている。

メニュー画面 9 1 上には、ユーザのリモートコントロールコマンドのキー操作などに応じて移動可能なカーソルが表示される。画像ファイル「200.png」の表示にカーソル表示が重なると、ファイル「000.html」において定義されるイベントMouseoverが発生される。このイベントMouseoverが発生されると、当該画像ファイル「200.png」にフォーカスが当たっていることを表現するために、イベントonMouseoverに対応するイベントハンドラonMoverhandler()が実行される。イベントハンドラonMoverhandler()の実行により、画像ファイル「200.png」が画像ファイル「201.png」に置き換えられる。画像ファイル「201

.png」は、画像ファイル「200.png」に対して配色が変えられたボタン画像などである。

カーソル表示が画像ファイル「201.png」上にあるときに、ユーザにより、リモートコントロールコマンドの所定のキー操作によりクリック操作が行われると、イベントonclickに対応するイベントハンドラonMclickhandler()が実行され、画像ファイル「201.png」が、選択されたことを表現する画像ファイル「202.png」に置き換えられる。画像ファイル「202.png」は、例えばボタンが押されたことを仮想的に表現するようなボタン画像である。

10    このように、「フォーカスされた」、「クリックされた」というイベント発生に対応するイベントハンドラをファイル「000.html」中に記述することで、ユーザの入力に反応するインタラクティブな機能を有するメニュー画面が実現される。

メニュー画面91において、「Scenario001」と描画されたボタン画像がクリックされると、シナリオ「Scenario001」の再生処理に移行される。ファイル「scenario001.js」が実行されることで、シナリオ「Scenario001」が再生される。第44図に示したように、ファイル「scenario001.js」中に記述されたメソッドplayPlaylist("00001.rpls")が呼び出され、プレイリスト  
20    「00001.rpls」が再生される。

プレイリスト「00001.rpls」の再生が終了されると、プレイリスト再生終了のイベントPlaylistEnded()が発生され、このイベントに対応したイベントハンドラplayScenario("scenario000.js")により、シナリオ「Scenario000.js」を再生するように指示される。この例では  
25    、シナリオ「Scenario001」の再生が終了されると、再びメニュー画面91が表示される。

シナリオ「Scenario001」の再生中に、「keyID」で指定されるキーが操作されても、シナリオ「Scenario000.js」の再生が指示され、再びメニュー画面91が表示される。

- 5 第41図～第44図で示したHTMLおよびECMASクリプトの記述は、一例であって、これはこの例に限定されるものではない。HTMLやECMASクリプトの記述には自由度があり、一部を異なる記述にしても同等の動作を実現するようにできる。

## 2-8. シンタクスについて

- 10 シナリオ記述言語として独自コマンドを用いた場合の、各ファイルのシンタクスについて説明する。まず、シナリオを記述するコマンドやデータベースのディスクへの格納方法について説明する。第45図は、シナリオ記述言語として独自コマンドを用いた場合の一例のファイルの管理構造を示す。
- 15 ディスク上には、まず、1つのルートディレクトリが作成される。このルートディレクトリの下が、1つの再生システムで管理される範囲とする。ルートディレクトリの下に、ディレクトリBDMVが置かれる。図示はされていないが、ディレクトリBDMVは、第9図の如く、ルートディレクトリの下に複数を置くことができる。
- 20 ディレクトリBDMVの下には、2つのファイルすなわちファイル「scenario.hdmv」およびファイル「entrylist.data」が置かれると共に、ディレクトリ「PLAYLIST」、ディレクトリ「CLIPINF」、ディレクトリ「STREAM」といった複数のディレクトリが置かれる。
- 25 第46図は、ファイル「scenario.hdmv」の一例の構造を表すシンタクスを示す。このファイル「scenario.hdmv」は、ディスク挿入時

などのイニシャルアクセス時に最初に読み込まれ実行されるファイルである。ファイル「scenario.hdmv」は、最初にファイル識別記号（フィールドtype\_indicator）、バージョン番号（フィールドversion\_number）が記述され、その後に、機能毎のデータを集めたブロックが  
5 並んでいる。

フィールドtype\_indicatorは、32ビットのデータ長を有し、このファイルがファイル「scenario.hdmv」であることを表す特定の文字列が格納されている。フィールドversion\_numberは、32ビットのデータ長を有し、バージョン番号が入る。フィールドScenario\_start\_addressは、32ビットのデータ長を有する符号無し of 整数で、ブロックScenario()が開始される位置をファイル「scenario.hdmv」の先頭からの相対バイト数で表した値が格納される。  
10

ブロックAutoplay()は、ファイルの先頭から41バイト目の固定位置から始まる。ブロックAutoplay()には、イニシャルアクセス時（ディスク挿入時のようにディスクの再生が始めて行われる時）に実行されるプログラムが記述される。Autoplay()の後には、任意の数のパディングワード（padding\_word）が入り、ブロックの後に空きを持たせることができるようになっている。  
15

第47図は、第46図におけるブロックAutoplay()の一例のデータ構造を表すシンタクスを示す。フィールドlengthは、データ長が32ビットの符号無し of 整数で、このフィールドlengthの直後からブロックAutoplay()の終わりまでのデータ長を、バイトで表したものが記述される。フィールドnumber\_of\_commandsは、その後に続くフィールドcommand(i)の数を表す。フィールドcommand(i)は、データ長が32ビットの値で、上述の第33図A、第33図B、第33図C、第33図D、第33図E、第33図F、第33図Gおよび第33図Hに一例が  
20  
25

記されている、プレーヤのパラメータのセット、指定のプレイリストの再生開始、演算などの命令が記述されている。

ブロックScenario()は、これまで説明してきた「シナリオ」が記述される。プレイリストの再生順序の情報およびプレイリスト毎に持つ  
5 ローカルコマンド領域などが、このブロックScenario()に記述される。

第48図は、ブロックScenario()の一例の構造を表すシンタクスを示す。ブロックScenario()は、シナリオの情報、すなわちプレイリスト間のリンクを定義するブロックであって、上述したプレコマンド、  
10 ポストコマンド、プレイアイテムコマンドなどの情報およびコマンドそのものが格納される。ブロックScenario()は、ブロックScenario()内に格納されるコマンドの情報を示すフィールドが記述される領域と、実際のプレコマンド、ポストコマンド、プレイアイテムコマンドが列挙される領域とから構成される。

15 フィールドlengthは、このフィールドlengthの直後からブロックScenario()の終わりまでの長さをバイトで表現した値が記述される。フィールドnumber\_of\_PlayListsは、シナリオを構成するプレイリストの数を表す。このフィールドnumber\_of\_PlayLists以降は、プレイリスト毎に持つデータである。すなわち、フィールドnumber\_of\_PlayLi  
20 stsの次行からのforループにより、フィールドnumber\_of\_PlayListsで示される値をループカウンタiの最大値として、プレイリスト毎のデータが記述される。

フィールドPre\_Command\_start\_idは、プレイリストを再生する前に実行されるプログラム（プレコマンド）の、コマンドテーブル中での  
25 開始番号を表す。フィールドPre\_Command\_start\_idにより示される番号は、後述するフィールドPL\_Command(i)が列挙されるforループ内で

のループカウンタ  $j$  を表す。同様に、フィールド `Post_Command_start_id` は、プレイリストを再生した後に実行されるプログラム（ポストコマンド」）の、コマンドテーブル中での開始番号を表す。フィールド `Post_Command_start_id` により示される番号は、後述するフィールド `PL_Command(j)` が列挙される `for` ループ内でのループカウンタ  $j$  を表す。

フィールド `number_of_Pre_Commands` は、このプレイリストを再生する前に実行されるプログラム（プレコマンド）を構成するコマンドの数を表す。同様に、フィールド `number_of_Post_Commands` は、このプレイリストを再生した後に実行されるプログラム（ポストコマンド）を構成するコマンドの数を表す。これらのプログラムは、後述するコマンドテーブル内に記述される。

フィールド `number_of_PlayItems` は、このプレイリストを構成するプレイアイテムの数を表す。フィールド `PI_Command_start_id` は、プレイアイテムの再生後に実行するコマンドの、コマンドテーブル中の開始番号を表す。フィールド `PI_Command_start_id` により示される番号は、後述するコマンドテーブル内でのループカウンタ  $j$  を表す。フィールド `number_of_PI_Commands` は、プレイアイテムの再生後に実行するコマンド（プレイアイテムコマンド）の数を表す。後述するコマンドテーブル内において、フィールド `PI_Command_start_id` で示される位置から始まり、フィールド `number_of_PI_Commands` で表される数のコマンドを、プレイアイテムの再生後に実行する。

フィールド `number_of_PL_Commands` は、次行からのコマンドテーブル中のコマンドの数を表す。コマンドテーブルは、フィールド `PL_Command(j)` が列挙される `for` ループにより構成される。コマンドテーブル内のコマンドには、順に番号  $j$  が割り当てられている。番号  $j$  は、

コマンドテーブルを記述するためのforループのループカウンタ  $j$  に対応する。フィールド `PL_Command(j)` は、一つのコマンドを表し、番号  $j$  は、上述したフィールド `Pre_Command_start_id`、フィールド `Post_Command_start_id` およびフィールド `PI_Command_start_id` から参照される。

第49図は、ファイル「entrylist.data」の一例のデータ構造を表したシンタクスを示す。ファイル「entrylist.data」には、最初にファイル識別記号（フィールド `type_indicator`）、バージョン番号（フィールド `version_number`）、ブロックの開始アドレス（フィールド `ScenarioEntry_start_address`）が記述され、その後、機能毎のデータを集めたブロックが並んでいる。

フィールド `type_indicator` には、データ長が32ビットを有し、このファイルがタイトルやメニューのエントリポイントを記述したファイルであることを表す特定の文字列が格納されている。フィールド `version_number` には、データ長が32ビットを有し、バージョン番号が格納される。フィールド `ScenarioEntry_start_address` には、データ長が32ビットの符号無しで、ブロック `ScenarioEntry()` が開始される位置を、ファイル「entrylist.data」の先頭からの相対バイト数で表した値が格納される。

第50図は、ブロック `AppInfo()` の一例の構造を表すシンタクスを示す。フィールド `length` には、データ長が32ビットの符号無しで、このフィールド `length` の直後からブロック `AppInfo()` の終わりまでの長さをバイトで表したものが記述される。フィールド `BDMV_name_character_set` は、後述するフィールド `BDMV_name` を記述する文字セットを表す。フィールド `PIN_valid_flag` は、再生時に暗証番号を設定するか否かを表し、設定有効の場合は、直後のフィールド `PIN` が暗証

番号を表す。フィールドBDMV\_name\_lengthは、次のフィールドBDMV\_nameの有効部分の長さを表す。フィールドBDMV\_nameは、このファイル「entrylist.data」が置かれるディレクトリBDMVに付けられた名称がテキスト形式で格納されている領域である。フィールドBDMV\_nameは、データ長が255バイトの固定長とされる。実際に名称が格納される部分は、フィールドBDMV\_nameの先頭からフィールドBDMV\_name\_lengthで指定されている長さまでとされる。

第51図は、ブロックScenarioEntry()の一例の構造を表すシンタックスを示す。ブロックScenarioEntry()は、シナリオ内の頭出し点が羅列される。シナリオは、上述したようにディレクトリBDMVに対して一つ作られ、ディレクトリBDMVの下に置かれる複数のプレイリストを結び付けて再生順序を定義する。ユーザから見た場合、シナリオは、必ずしも一つの映像・音声単位として見える訳ではなく、複数の「タイトル」から構成されているように見える。

例えば、一枚のディスクに映画が3本収録されている場合、再生順序を定義するシナリオは、ディスクに一つのみ存在するが、ユーザには当該ディスクに3つのタイトルが収録されているように見える。あるいは、その3つのタイトルのリストを表示し、ユーザに選択させるタイトルメニュー画面も含めて、4つのタイトルに分割されているように見える。メニュー画面もユーザにとっては一まとまりの映像、音声単位であるので、この発明の実施の一形態では、タイトルの一種としてみなすことにする。

このように、プレイリストの結びつきを定義するシナリオという単位と、ユーザが一まとまりの映像、音声として認識する単位には違いがあるため、シナリオ内に頭出し点を定義しておく必要がある。このシナリオ内の頭出し点を、タイトルエントリと称する。このブロック



ScenarioEntry()に、タイトルエントリの情報が記述される。

第51図の説明に戻り、フィールドlengthは、データ長が32ビットの符号無しで、このフィールドlengthの直後からブロックScenarioEntry()の終わりまでの長さをバイトで表したものが格納される。フィールドname\_character\_setは、後述するフィールドTopMenu\_nameとフィールドTitle\_nameとを表現する文字セットを表す。

次のブロックTop Menu PL()は、ユーザがリモコンのタイトルメニューキーを押したときに表示されるメニューを構成しているプレイリストあるいはプレイリスト群へのエントリポイントを指定する。トップメニューは、シナリオに一つあり、例えばタイトルをユーザに提示するためのメニューとして利用される。トップメニューに対して、さらにオーディオや字幕を設定するサブメニューを下層メニューとして設けることができる。サブメニューは、ストリーム設定メニューとも称される。

フィールドflagsは、詳細は省略するが、トップメニューに関する属性情報を表す領域である。フィールドTopMenu\_ref\_to\_PlayList\_file\_nameは、トップメニューを構成するプレイリストあるいはプレイリスト群の入り口となるプレイリストを指定する。フィールドTopMenu\_ref\_to\_PlayItem\_idは、フィールドTopMenu\_ref\_to\_PlayList\_file\_nameで指定したプレイリスト中の特定のプレイアイテムからトップメニューが開始されている場合、そのプレイアイテムの番号を指定する。プレイリストの先頭からの再生であれば、フィールドTopMenu\_ref\_to\_PlayItem\_idの値は、「0」となる。フィールドTopMenu\_name\_lengthは、トップメニューに付された名前の長さを表す。フィールドTopMenu\_nameは、トップメニューに付された名前の文字列が格納される。

ブロックTop Menu PL()の次に、タイトルに関する情報が記述される。フィールドnumber\_of\_Titlesは、その直後のforループ内のタイトル頭出し点（タイトルエントリ）の数を表す。フィールドflagsは、詳細は省略するが、タイトルに関する属性情報を表す領域である。

- 5   フィールドTitle\_ref\_to\_PlayList\_file\_nameは、タイトルエントリを含むプレイリストのファイル名を表す。フィールドTitle\_ref\_to\_PlayItem\_idは、フィールドTitle\_ref\_to\_PlayList\_file\_nameで指定したプレイリストの中の特定のプレイアイテムからタイトルが始まる場合に用いる。フィールドTitle\_name\_lengthは、タイトルに付された名前
- 10   の長さを表す。フィールドTitle\_nameは、タイトルに付された名前の文字列が格納される。

サブメニューに関する情報が記述される。「Stream Setup Menu」以下に、プレイアイテム毎に持つことができるストリーム設定メニュー（すなわちサブメニュー）を構成するプレイリストあるいはプレイ

15   リスト群へのエントリポイントが記述されている。ストリーム設定メニューは、オーディオ、字幕、アングルなどの切り替えなど、プレイリスト毎に個別にメニューを用意したい場合に利用することができる。例えば、上述の第25図におけるボタン64、65を押すことで表示される画面がサブメニューである。

- 20   フィールドnumber\_of\_PlayListsは、ストリーム設定メニューに用いられるプレイリストの数を表す。このフィールドnumber\_of\_PlayListsの値が直後のforループのループ回数として用いられる。フィールドSSMenu\_flagsは、詳細は省略するが、ストリーム設定メニューに関連する属性情報を格納する領域である。フィールドSSMenu\_ref\_to\_PlayList\_file\_nameは、ストリーム設定メニューを構成するプレイリス
- 25   トあるいはプレイリスト群の入り口となるプレイリストを指定する。

フィールドSSMenu\_ref\_to\_PlayItem\_idは、フィールドSSMenu\_ref\_to\_PlayList\_file\_nameで指定したプレイリスト中の特定のプレイアイテムからストリーム設定メニューが開始されている場合、そのプレイアイテムの番号を指定する。プレイリストの先頭からの再生であれば、フィールドSSMenu\_ref\_to\_PlayItem\_idの値は、「0」となる。

第52図は、ファイル「xxxxx.mpls」の一例の構造を表すシンタックスを示す。第52図において、ファイル「xxxxx.mpls」の内部は、機能別の情報毎にブロックが構成される。フィールドtype\_indicatorにこのファイルを示す文字列が格納され、フィールドversion\_numberにこのファイルのバージョンが示される。フィールドPlayList\_start\_addressおよびPlayListMark\_start\_addressには、それぞれ対応するブロックの先頭アドレスが例えばデータ長が32ビットのアドレス情報として示される。

ブロックPLControlInfo()は、このプレイリストに関する属性情報が格納される。ブロックPlayList()は、このプレイリストを構成するプレイアイテムに関する情報が格納される。ブロックPlayListMark()は、このプレイリストに付されるマークの情報が格納される。

ファイル「xxxxx.mpls」において、ブロックPLControlInfo()、PlayList()およびPlayListMark()の先頭アドレスがこれらのブロックより前に示されるために、各ブロックの前および／または後ろに、任意の長さのパディングデータpadding\_wordを挿入することができる。ファイル「xxxxx.mpls」の最初のブロックであるブロックPLControlInfo()の開始位置は、ファイルの先頭から41バイト目に固定される。

第53図は、ブロックPLControlInfo()の一例の構造を表すシンタックスを示す。このブロックPLControlInfo()は、プレイリストの再生において直接的に必要とされない、プレイリストに関する各種の属性

情報が格納される。フィールド`PlayList_character_set`は、プレイリストに関する文字列情報の文字セットが指定される。

フィールド`PL_playback_type`は、第54図に一例が示されるような値を取り、このプレイリストがシーケンシャルに再生される通常のプレイリストか、プレイアイテムをランダム再生するプレイリストか、プレイアイテムをシャッフル再生するプレイリストであるかを表す。ランダムシャッフルは、プレイリストの単位で指定する。一つのプレイリストに通常再生のプレイアイテムとランダムシャッフルのプレイアイテムブロックを混在させない。再生専用の記録媒体の場合には、  
10 制作者の意図により、ランダム再生やシャッフル再生を指定されることがあるため、このような情報が必要とされる。

フィールド`playback_count`は、このプレイリストがランダム再生あるいはシャッフル再生のプレイリストであるとき、プレイアイテムの再生回数を指定する。このフィールド`playback_count`により、ランダム再生あるいはシャッフル再生に用いるプレイアイテムの数を指定で  
15 きる。

フィールド`PL_UOP_mask_table()`は、ユーザ操作を制限する情報が格納される。再生、早送り、早戻しなどのユーザ操作を、このプレイリストの再生中に禁止したい場合、この領域で適切な指定をする。例えば、このフィールド`PL_UOP_mask_table()`内の値を所定に設定することで、警告表示や著作権の表示を早送りなどで飛ばして再生されないようにできる。  
20

フィールド`PL_random_access_mode`は、第55図に一例が示される値を取り、他プレイリストからこのプレイリスト中の任意の箇所に飛び込み再生をするランダムアクセスが可能か否かを示す。例えば、必ずユーザに見せたいプレイリストがある場合に、フィールド`PL_rando`  
25

m\_access\_modeの値を〔0 x 1〕に設定する。このプレイリストに飛び込んで再生する際には、早送り、早戻し、任意時刻からの再生などが不可能になり、必ずプレイリストの先頭からの再生となる。再生専用の記録媒体においては、制作会社のロゴ表示や注意事項など、ユーザに必ず見せたいシーンが収録されている場合がある。このフィールドPL\_random\_access\_modeは、そのようなシーンを変速再生等で飛ばされないようにするために必要な情報である。フィールドPL\_UOP\_mask\_table()が当該プレイリストの再生中の操作の制限であるのに対し、このPL\_random\_access\_modeは他プレイリストから当該プレイリストに飛び込んで再生を開始する際の制限を定めているという違いがある。

フィールドPlayList\_durationは、プレイリストの再生時間を示す。フィールドPlayList\_nameは、フィールドPlayList\_name\_lengthに示される値を有効長としてプレイリスト名が示される。フィールドPlayList\_detailは、フィールドPlayList\_detail\_lengthに示される値を有効長としてプレイリストの詳細情報が示される。

第56図は、ブロックPlayList()の一例の構造を表すシンタクスを示す。フィールドlengthは、フィールドlengthの直後のフィールドからこのブロックPlayList()の終端までのバイト長を示す。フィールドnumber\_of\_PlayItemsは、このプレイリストを構成するプレイアイテムの数を示す。フィールドnumber\_of\_SubPlayItemsは、このメインのプレイアイテムと同時に再生される補助的なプレイアイテム（サブプレイアイテム）の数を示す。

ブロックPlayItem()は、プレイアイテムの情報が格納される。ブロックSubPlayItem()は、サブプレイアイテムの情報が格納される。

第57図は、ブロックPlayItem()の一例の構造を表すシンタクスを

示す。フィールドClip\_Information\_file\_nameは、このプレイアイテムが参照しているクリップと1対1に対応するクリップ情報ファイル（拡張子が「clpi」であるファイル）のファイル名が文字列で示される。

- 5      フィールドClip\_codec\_identifierは、このプレイアイテムにより参照されるクリップの符号化方式を示す。この実施の一形態では、フィールドClip\_codec\_Identifierは、値「M 2 T S」に固定的とされる。すなわち、この実施の一形態では、プレイアイテムにより参照されるクリップの符号化方式が値「M 2 T S」により示される方式に固定
- 10    定的とされる。

フラグis\_multi\_angleは、このプレイアイテムがマルチアングル構造であるかどうかを表す。

- フィールドconnection\_conditionは、このプレイアイテムと次のプレイアイテムとがどのように接続されているかを示す情報である。フ
- 15    ィールドconnection\_conditionにより、プレイアイテムとプレイアイテムとの間が継ぎ目なくシームレスに再生できるか否かが示される。

- フィールドref\_to\_STC\_idは、このプレイアイテムにより参照されるクリップ内のシーケンスSTC\_sequenceを指定する。シーケンスSTC\_sequenceは、M P E G 2    T S (Transport Stream)における時間軸の
- 20    基準であるP C R (Program Clock Reference)が連続な範囲を表すB l u - r a y    D i s c規格独自の構造である。シーケンスSTC\_sequenceには、クリップ内で一意な番号STC\_idが割り当てられる。このシーケンスSTC\_sequence内では、不連続の無い一貫した時間軸を定義できるので、プレイアイテムの開始時刻および終了時刻を一意に定める
- 25    ことができる。つまり、各プレイアイテムの開始点と終了点は、同一のシーケンスSTC\_sequenceに存在していなければならない。フィール

ドref\_to\_STC\_idでは、番号STC\_idによりシーケンスSTC\_sequenceが指定される。

- フィールドIN\_timeおよびOUT\_Timeは、このプレイアイテムにおける開始点および終了点の、シーケンスSTC\_sequence 上でのタイムスタンプpts(presentation\_time\_stamp)をそれぞれ示す。

- フィールドPI\_UOP\_mask\_table()は、ユーザ操作を制限するためのデータが格納される。ここで制限されたユーザ操作は、仮にユーザがその操作を行ったとしても、プレーヤは反応してはならない。例えばメニュー画面で早送りが実行されないようにする場合に、ここで早送りのユーザ操作を無効にするような設定をしておく。

- このプレイアイテム毎に設けられるフィールドPI\_UOP\_mask\_table()は、上述したプレイリストの再生に関するブロックPLControlInfo()に設けられるフィールドPL\_UOP\_mask\_table()と同様の目的の情報が格納される。プレイリストとプレイアイテムとの何方かで禁止となっていれば、そのユーザ操作は禁止となる。プレイリストの情報と、プレイアイテムの情報とで和演算がなされ、あるプレイアイテム再生中でのユーザ操作の制限が決まることになる。

フィールドPID\_filter()は、詳細は省略するが、このプレイアイテムで再生するストリームの優先順位を決めるテーブルである。

- フィールドPI\_random\_access\_modeは、第58図に一例が示されるような値を取り、このプレイアイテム中の任意の箇所に飛び込み再生をするランダムアクセスが可能か否かを示す。例えば、必ずユーザに見せたいプレイリストがある場合に、フィールドPI\_random\_access\_modeの値を「0 x 1」に設定しておく。これにより、このプレイアイテムを再生開始する際に、早送り、早戻し、任意時刻からの再生などを不可能にできる。

フィールドstill\_modeは、このプレイアイテム再生後に一時停止を行うかを定めるフィールドである。フィールドstill\_modeは、第59図に一例が示されるような値を取り、例えばフィールドstill\_modeの値が[0 x 1]のときは一時停止することを表し、次のフィールドstill\_timeのフィールドで指定された時間、一時停止を行う。これにより、スライドショーなどのような、静止画像を一定間隔で次々と表示するような再生を実現することができる。この場合には、静止画1枚ずつがそれぞれプレイアイテムとなっている。フィールドstill\_timeは、有限の時間だけでなく、ユーザが入力するまで一時停止し続けるという設定（ポーズ設定）も可能となっている。例えばフィールドstill\_modeの値を[0 x 2]とすることで、ポーズ設定が可能とされる。

上述したフラグis\_multi\_angleの値が例えば「1」とされていれば、このプレイアイテムがマルチアングルであるとされ、「Angle」以下に、マルチアングルのための情報が追加される。

フィールドnumber\_of\_anglesは、アングル数を表す。フィールドis\_seamless\_angle\_changeは、第60図に一例が示されるような値を取り、アングルがシームレスに切り替え可能なように、すなわち、アングル間をなめらかに切り替えることができる状態に各アングルがディスク上配置されているか否かを表す。

次のforループは、アングルを構成するクリップの情報が記述される。forループ中のフィールドClip\_Information\_file\_nameは、このプレイアイテムが参照しているクリップと1対1に対応するクリップ情報ファイル（拡張子が「clpi」であるファイル）のファイル名が文字列で示される。フィールドref\_to\_STC\_idは、このプレイアイテムにより参照されるクリップ内のシーケンスSTC\_sequenceを指定する。



値angle\_id=0に相当するアングルは、アングルを構成しない通常のプレイアイテムと同じように、このブロックPlayItem()の前半部分で既に定義されている。値angle\_id=1以降のアングルがこのforループの中で定義されている。このforループの中には、値angle=0に相当するアングルは、含まれていない。

第61図は、ブロックSubPlayItem()の一例の構造を表すシンタクスを示す。フィールドlengthは、この直後のフィールドからSubPlayItem()の終わりまでの長さをバイトで表したものである。フィールドClip\_Information\_file\_nameは、このサブプレイアイテムが参照しているクリップと1対1に対応するクリップ情報ファイル（拡張子が「clpi」であるファイル）のファイル名が文字列で示される。

フィールドClip\_codec\_identifierは、このサブプレイアイテムにより参照されるクリップの符号化方式を示す。この実施の一形態では、フィールドClip\_codec\_Identifierは、値「M2TS」に固定的とされる。

フィールドis\_repeat\_flagは、第62図に一例が示されるような値を取り、このサブプレイアイテムをメインのプレイアイテム（メインパス）と非同期に繰り返し再生するか否かを表すフラグである。例えば、このフィールドis\_repeat\_flagの値が「1」の場合は、メインのプレイアイテムの再生が終了するまで、メインのプレイアイテムとは非同期に、このサブプレイアイテムが繰り返し再生される。このフィールドis\_repeat\_flagの値が「0」の場合は、このサブプレイアイテムは、メインのプレイアイテムと同期して、一度だけ再生される。

例えば、このサブプレイアイテムがオーディオのみのサブプレイアイテムである場合、フィールドis\_repeat\_flagの値を「1」とすることで、BGM(Back Ground Music)再生を行うことができる。

フィールドSubPlayItem\_typeは、このサブプレイアイテムがどのような性質を持ったサブプレイアイテムであるかを表す。例えば、フィールドSubPlayItem\_typeの値が「1」であれば、オーディオのみのサブプレイアイテムであることを表す。

- 5     フィールドref\_to\_STC\_idは、このプレイアイテムにより参照されるクリップ内のシーケンスSTC\_sequenceを指定する。フィールドSubPlayItem\_IN\_timeおよびSubPlayItem\_OUT\_Timeは、このサブプレイアイテムにおける開始点および終了点の、シーケンスSTC\_sequence 上でのタイムスタンプpts(presentation\_time\_stamp)をそれぞれ示す。
- 10    上述のフィールドis\_repeat\_flagの値が例えば「0」であって、このサブプレイアイテムがメインのプレイアイテムと同期して再生されることを示す場合、フィールドsync\_PlayItem\_idとフィールドsync\_start\_PTS\_of\_PlayItemにより、サブプレイアイテムがメインのプレイアイテムのどの時刻から同期再生されるかが指定される。
- 15    第63図に一例が示されるように、フィールドsync\_PlayItem\_idで、メインパスのプレイアイテムを指定し(PlayItem=1)、フィールドsync\_start\_PTS\_of\_PlayItemで、サブプレイアイテムが再生開始されるメインのプレイアイテム上の時刻を表す(t1)。サブプレイアイテムとして、対応するクリップがフィールドSubPlayItem\_IN\_timeおよびフィールドSubPlayItem\_OUT\_timeで指定される期間、再生されることが示されている。
- 20

- 第64図は、ファイル「zzzzz.clpi」の一例の構造を表すシンタクスを示す。第53図において、ファイル「zzzzz.clpi」の内部は、機能別の情報毎にブロックが構成される。フィールドtype\_indicatorにこのファイルを示す文字列が格納され、フィールドversion\_numberにこのファイルのバージョンが示される。フィールドSequenceInfo\_sta
- 25

rt\_address、フィールドProgramInfo\_start\_address、フィールドCPI\_start\_addressおよびフィールドClipMark\_start\_addressで、それぞれ対応するブロックの開始位置が示される。

第65図は、ブロックClipInfo()の一例の構造を表すシンタクスを示す。フィールドlengthは、この直後のフィールドからブロックClipInfo()の終わりまでの長さをバイトで表す。

フィールドapplication\_typeは、クリップAVストリーム（拡張子が「m2ts」のファイル）がどのような多重化によって作られているかを示す。フィールドapplication\_typeは、第66図に一例が示されるような値を取り、このクリップAVストリームが通常のビデオストリームであるか、静止画の表示に適した多重化がなされているストリームであるかを表す。

より具体的には、この例によれば、フィールドapplication\_typeの値が「1」で、対応するクリップAVストリームのファイルがこの実施の一形態によるBDMVトランスポートストリームのルールに従っていることを示す。当該クリップAVストリームは、通常の動画が再生される。

フィールドapplication\_typeの値が「2」で、対応するクリップAVストリームのファイルがオーディオ再生に同期する静止画用の、BDMVトランスポートストリームのルールに従っていることを示す。当該クリップAVストリームは、例えばMP EG 2形式のファイルであって、ビデオデータおよびオーディオデータがマルチプレクスされている。ビデオデータは、例えばMP EG 2におけるIピクチャが静止画として並んでいるような構成とされる。これにより、例えばオーディオの時間軸に同期して表示されるスライドショーのような再生が可能とされる。このような再生を、タイムベーススライドショーと称

する。

フィールドapplication\_typeの値が「3」で、対応するクリップAVストリームのファイルがオーディオとは非同期に再生される静止画用の、BDMVトランスポートストリームのルールに従っていることを示す。オーディオデータとビデオデータとは、別ファイルとされ、例えば、オーディオデータが再生されている間、ビデオデータは、静止画が任意の間隔またはユーザの指定に基づき切り換えられて表示される。ビデオデータは、例えばMPEG2におけるIピクチャが静止画として並んでいるような構成とすることができる。このような再生を、ブラウザブルスライドショーと称する。

フィールドapplication\_typeの値が「0」のときは、対応するクリップAVストリームがBDMVトランスポートストリームのルールに従っていない場合である。

静止画の表示に適した多重化とは、主として静止画スライドショーのようなアプリケーションの実現を容易にすることを想定している。このようなアプリケーションにおいては、静止画1枚と、その上に重ねる字幕やグラフィクスデータとをカプセル化して多重化すると、読み込みが容易となる。

通常の動画像と同様の多重化をしてしまうと、ある静止画と同時に表示されるべき字幕が、前の静止画の画像データ付近に多重化されているような状況（いわゆる多重化位相差）が発生し、より広範囲のストリームデータを読みださないと字幕とグラフィクスとが重ねられた静止画像1枚を表示することができない。

この発明の実施の一形態においては、ビデオデータや字幕を表示するためのグラフィクスデータは、MPEG-2システム規格におけるTS (Transport Stream) パケットに格納されて伝送される。1つのT

S パケットは、188 バイトからなり、上述のビデオデータやグラフィクスデータは、それぞれTS パケットに収まるように分割されてTS パケットに格納される。ある静止画像データ（画像P1とする）に対応した字幕データの packets が、次以降の静止画像データ（画像P2とする）の packets の後ろに配置されてしまうと、画像P1に対応した字幕を表示するために、画像P2の後ろまでのデータを読み込んでおかなければならない。

ある静止画一枚とそれに付随する字幕およびグラフィクスだけを対象として多重化を行えば（カプセル化）、他のデータが混入しないストリームを作ることができる。それを静止画ごとに繰り返してストリームをつなげていくと、静止画（および当該静止画に付随する字幕、グラフィクスデータ）毎のデータが直列につながった一本のストリームが出来る。このような多重化を行ったストリームを、静止画用のBDMVストリームとしている。

15 静止画用のBDMVストリームには、タイムベーススライドショーと、ブラウザブルスライドショーとの2種類がある。この実施の一形態では、この2種類を、フィールドapplication\_typeで別の番号を割り当てて、区別している。

20 静止画とそれに付随する字幕やグラフィクスとをカプセル化して記録することで、静止画を切り替えながら再生する際のアクセス性が向上する。

第65図の説明に戻り、フィールドClip\_stream\_typeは、クリップAVストリームの種別を表す。再生専用の規格においては、通常のクリップであることを示す値、例えば値「1」に固定的としてよい。フィールドTS\_recording\_rateは、クリップAVストリームファイルの記録レートをバイト/秒で表したものである。フィールドnum\_of\_sou

rce\_packetsは、クリップAVストリームに含まれるパケット数を表す。フィールドBD\_system\_useおよびブロックTS\_type\_info\_block()は、この発明と関連が低いので説明を省略する。

第67図は、ブロックSequenceInfo()の一例の構造を表すシンタックスを示す。フィールドlengthは、この直後のフィールドからブロックSequenceInfo()の終わりまでの長さをバイトで表したものである。フィールドnum\_of\_ATC\_sequencesは、連続した時間に記録されたことを表すシーケンスATC\_sequenceの数を示す。再生専用の媒体の場合、シーケンスATC\_sequenceの数は、「1」にすることができるため、詳細は省略する。フィールドSPN\_ATC\_startは、シーケンスATC\_sequenceの開始をパケット番号で表したものであり、シーケンスATC\_sequenceが1つの場合には、クリップAVストリームファイルの先頭と一致し、値が「0」となる。

フィールドnum\_of\_STC\_sequencesは、シーケンスATC\_sequence上のシーケンスSTC\_sequenceの数を表す。再生専用の媒体の場合、シーケンスSTC\_sequenceの数は、「1」にすることができるため、詳細は省略する。フィールドoffset\_STC\_idは、値が「0」に固定的とされる。フィールドPCR\_PIDは、MPEG2 TSにおいて、PCR(Program Clock Reference)が含まれるTSパケットのPIDを表す。フィールドSPN\_STC\_startは、シーケンスSTC\_sequenceの開始をパケット番号で表したものであり、シーケンスSTC\_sequenceが1つの場合には、クリップAVストリームファイルの先頭と一致し、値が「0」となる。フィールドpresentation\_start\_timeおよびフィールドpresentation\_end\_timeは、クリップAVストリーム中の有効な範囲を表す。フィールドpresentation\_start\_timeおよびフィールドpresentation\_end\_timeで示される範囲がプレイアイテムから参照できる範囲となる。

第68図は、ブロックProgramInfo()の一例の構造を表すシンタックスを示す。記録再生媒体用のシンタックス構造を再生専用媒体に適用することができ、新規の構造は無いので詳細な説明は省略する。フィールドnum\_of\_program\_sequencesの値が「1」であること、フィールドnum\_of\_groupsの値が「1」であるなどの制限を加えることができる。

第69図は、ブロックStreamCodingInfo()一例の構造を表すシンタックスを示す。ブロックProgramInfo()と同様、記録再生媒体用のシンタックス構造と略同じ構造を有し、ビデオデータに関しては、ビデオデータのフォーマット、フレームレート、アスペクト比などの属性情報、オーディオデータに関しては、サンプリング周波数などの属性情報が記述される。再生専用媒体に適用する際には、この第69図に示されるように、字幕、オーディオストリームの言語を表すフィールドlanguage\_codeを追加する必要がある。この情報は、プレーヤの言語設定に従って最適な音声・字幕の言語を選択する際に有用である。

第70図は、ブロックCPI()の一例の構造を表すシンタックスを示す。一般に、MPEGストリームのような、フレーム間圧縮を行っている符号化ストリームにおいては、デコード開始可能な箇所は、GOP(Group Of Picture)の先頭など一部の箇所に限定されている。CPI(Characteristic Point Information)とは、そのデコード可能な開始点の位置の情報を集めたデータベースで、再生時刻とファイル内アドレスとが対応付けられたテーブルになっている。CPIは、デコード単位の先頭位置を示す情報がテーブル化されている。

このようにデータベースを定めることで、例えば、任意の時刻から再生したい場合、再生時刻を元にCPIを参照することによって再生位置のファイル内アドレスがわかる。このアドレスは、デコード単位

の先頭となっているため、プレーヤは、そこからデータを読み出してデコードし、素早く画像を表示することができる。

このC P Iに格納される、デコード単位の先頭位置（この例ではG O Pの先頭位置）を、E P (Entry Point) エントリと称する。

- 5     フィールドCPI\_typeは、C P Iの種類を表し、第71図に一例が示されるような値を取る。この発明の場合は、再生専用媒体用のC P Iであることを示す値となる。具体的には、値が「8」とされ、B D M V用のE P エントリのマップ(EP\_map\_type\_for\_BDMV)が格納されることが示される。
- 10    第72図は、再生専用媒体用のE P エントリのマップEP\_map、すなわち、上述のフィールドCPI\_type内のブロックEP\_map\_for\_BDMV()の一例のデータ構造を表すシンタクスを示す。マップEP\_mapは、G O Pの先頭位置について、再生時刻とファイル内アドレスを対応付けたテーブルである。第72図の例では、G O Pの先頭位置について、M P
- 15    E GのP T S (Presentation Time Stamp)とS P N (Source Packet Number)とが対応付けられたテーブルとしてデータベースが作られている。S P Nは、ソースパケット番号を示し、ファイルの先頭からのバイトアドレスに相当する。

- 記録再生用のマップEP\_mapの構造と、この再生専用媒体用のマップ
- 20    EP\_mapの構造は、略同一とされている。この実施の一形態では、データ量の削減および検索の高速化のために、各値をcoarseとfineに分割し、大まかな単位での検索(coarse)と、より精密な単位での検索(fine)とを分けて行うことが可能なようにされている。そのため、内部構造がcoarseおよびfineのそれぞれに対応した二つのforループに分か
- 25    れており、「GOPの最初のIピクチャのPTS 対 ファイル内アドレス」という単純なテーブルと比べて、多少複雑になっている。



フィールドEP\_fine\_table\_start\_addressは、より精密な検索を行うためのテーブルの位置が示される。次のforループは、フィールドPTS\_EP\_coarseおよびSPN\_EP\_coarseにより、大まかな単位で検索を行う(coarse)ためのテーブルが格納され、フィールドref\_to\_EP\_fine\_idにより、大まかな単位から参照される精密な検索(fine)のためのテーブルの番号が記述される。フィールドPTS\_EP\_coarseおよびSPN\_EP\_coarseは、それぞれPTSおよびSPNの上位側のビットが示される。

次に、パディングワードを挟んで、精密な検索を行うためのフィールドPTS\_EP\_fineおよびSPN\_EP\_fineが格納されるforループが配される。それと共に、フラグis\_angle\_change\_pointおよびフィールドI\_end\_position\_offsetが格納される。フラグis\_angle\_change\_pointは、クリップAVストリームがマルチアングルを構成している場合に、各EPエントリがアングル切り替え可能点に該当しているか否かを示す。

## 2-9. デコーダモデル

第73図A、第73図Bおよび第73図Cは、この発明の実施の一形態に適用できるプレーヤデコーダ100の一例の構成を示す機能ブロック図である。このプレーヤデコーダ100は、図示されないドライブ装置に装填されたディスクから再生されたデータを解釈し、AVストリームを出力すると共に、出力されたAVストリームに対するユーザによるインタラクティブな操作を可能とする。

プレーヤデコーダ100は、図示されないCPU(Central Processing Unit)により全体の動作が制御される。例えば、プレーヤデコーダ100の各部におけるストリームやデータの流れは、CPUにより

監視され、制御される。

以下の説明では、特に記載がない限り、独自コマンドを用いて記述されたシナリオがプレーヤデコーダ100で実行されるものとする。

図示されないドライブ装置にディスクが装填されると、上述したよう  
5 うに、先ずファイル「scenario.hdmv」およびファイル「entrylist.data」が再生され、このファイル「scenario.hdmv」およびファイル「entrylist.data」の記述に基づき、必要な他のファイルが読み出され、ディスクに記録されたコンテンツが再生される。例えば、ファイル「scenario.hdmv」およびファイル「entrylist.data」の記述に基づ  
10 き、動画プレーン10に表示するための動画データ、字幕プレーン11やグラフィクスプレーン12に表示するための画像データ、プレイリストファイルなどがディスクから読み出される。

以下では、ディスクから読み出されるこれらのデータのうち、動画データ、サブピクチャ（字幕データ）や音声データといった、連続的に処理する必要があるストリームをリアルタイムストリームと称する  
15 。そして、シナリオファイル、プレイリストファイルといった、連続的な処理を要求されない非リアルタイムなデータを、ストアオブジェクトと称する。ストアオブジェクトは、メモリ上などに蓄積、展開され、必要に応じて処理される。

20 プレーヤデコーダ100は、チャンネル（1）および（2）の2系統の入力チャンネルを有し、入力チャンネル（1）の入力端101に、ストアオブジェクトが入力される。入力チャンネル（2）の入力端202に、リアルタイムストリームが入力される。入力端202に、ストアオブジェクトを入力することも可能である。この実施の一形態  
25 では、入力端202に入力されるリアルタイムストリームおよび一部のストアオブジェクトは、例えばMPEG2 TSである。

入力端 101 に入力されるストアオブジェクトは、ディスクから読み出されたデータに限られない。例えば、プレーヤデコーダ 100 に対してインターネットなどのネットワーク接続機能を持たせ、ネットワークを介して取得されたストアオブジェクトを入力端 101 に対して入力するようにしてもよい。ボタン画像を表示させるための画像データや新たなシナリオデータなどを、ネットワークを介して取得し、入力端 101 から入力することが考えられる。これに限らず、字幕データなどリアルタイムストリームとして扱われるデータをネットワークを介して取得し、入力端 101 から入力することも可能である。

- 10 入力端 202 に入力されるリアルタイムストリームは、MPEG 2 TS に限られない。パケット単位で伝送され、ビデオデータ、オーディオデータ、静止画像データなどを多重化可能であれば、他の形式のストリームを入力するようにしてもよい。このときには、後述する PID フィルタ 110 は、そのストリーム形式に適合したデマルチプレクサとして用いられ、ビデオデータ、オーディオデータ、静止画像データなどを分離する。

ドライブ装置においてディスクの回転速度を 2 倍速などの高速回転としてディスクからの読み出し転送レートを上げ、時分割で動作させることにより、ディスクからの、チャンネル (1) および (2) の 20 系統の読み出しが実現可能である。

入力チャンネル (1) の系統について説明する。入力端 101 に入力されたストアオブジェクトは、スイッチ回路 102 に入力される。ストアオブジェクトとして ECMA スクリプトや HTML ファイルなどによるプログラムコードが入力された場合、スイッチ回路 102 において出力端 102A が選択され、入力されたプログラムコードがコードバッファ 104 に蓄えられる。

ストアオブジェクトとして画像データが入力された場合、スイッチ回路102において出力端102Bが選択され、入力された画像データがスイッチ回路103に入力される。入力端202に入力されたりアルタイムストリームに、字幕プレーン11やグラフィクスプレーン12に表示するための画像データが含まれていない場合には、スイッチ回路103で入力端103Aが選択され、スイッチ回路102から入力された画像データがコンテンツバッファ105に蓄えられる。

同様にして、入力端202に入力されたりアルタイムストリームに、字幕プレーン11やグラフィクスプレーン12に表示するための画像データが含まれている場合には、スイッチ回路103において入力端103Bが選択され、当該画像データがコンテンツバッファ105に蓄えられる。コードバッファ104およびコンテンツバッファ105に蓄えられたストアオブジェクトは、必要に応じて読み出され、マルチメディアエンジン106に供給される。

コンテンツバッファ105に蓄えられたストアオブジェクトのうち画像データは、スイッチ回路107および108をそれぞれ介して、グラフィクスデコーダA116およびグラフィクスデコーダB117にも供給される。

マルチメディアエンジン106は、XMLパーサ106A、スクリプトインタプリタ106Bおよびグラフィクスレンダラ106Cを含む。マルチメディアエンジン106は、独立的なハードウェアで構成してもよいし、上述した図示されないCPUの、所定のプログラムに基づく処理で実現することも可能である。

XMLパーサ106Aは、XML (Extensible Markup Language) 文書を解析する機能を有し、HTML文書の解析も可能である。XMLパーサ106Aで解釈されたHTML文書は、このプレーヤデコーダ

100で実行可能な形式に変換される。スクリプトインタプリタ106Bは、ECMASクリプトを解析し、このプレーヤデコーダ100で実行可能な形式に変換される。グラフィクスレンダラ106Cは、画像データを、字幕プレーン11およびグラフィクスプレーン12に展開可能な形式にデコードする。

マルチメディアエンジン106において、バッファ109をワークメモリとして、これらXMLパーサ106A、スクリプトインタプリタ106Bおよびグラフィクスレンダラ106Cの処理が行われる。例えば、XMLパーサ106Aおよびスクリプトインタプリタ106Bにより、バッファ109のうちコードバッファ109Aが用いられる。グラフィクスレンダラ106Cにより、バッファ109のうちグラフィクスバッファ109Dが用いられる。バッファ109は、上述のコードバッファ109Aおよびグラフィクスバッファ109Dの他に、文字列の表示に用いるフォントデータが格納されるフォントバッファ109B、XMLパーサ106AでHTML文書を解析した結果を階層化された木構造で保持するためのツリーバッファ109Cなどが含まれる。

マルチメディアエンジン106では、例えば、シナリオ記述言語としてHTMLとECMASクリプトとを組み合わせ用いている場合コードバッファ104に蓄えられたECMASクリプトを読み出し、読み出されたECMASクリプトの記述に基づいて用いられる。さらに必要に応じて、マルチメディアエンジン106は、コードバッファ104からの他のECMASクリプトやHTML文書の読み出し、コンテンツバッファ105からの画像データの読み出しなどを行う。コードバッファ104およびコンテンツバッファ105に格納されたデータは、当該データが不要になるまで、コードバッファ104やコン

テンツバッファ105に保持しておくことができる。したがって、これらコードバッファ104やコンテンツバッファ105に格納されたデータは、必要に応じて何度でも読み出して使うことができる。

マルチメディアエンジン106では、上述の他にも、入力された複数種類のデータのデマルチプレクス処理、Java VM (Java (登録商標) 仮想マシン) 機能などが行われる。さらに、マルチメディアエンジン106により、ユーザからの、リモートコントロールコマンドやポインティングデバイスなどによる入力を受け取られ、それらの入力に基づく処理が行われる。ユーザ入力は、後述するグラフィックスデコードA116、グラフィックスデコードB117、オーディオデコード118、MPEGビデオデコード120およびシステムデコード121にも供給される。

グラフィックスレンダラ106Cで処理された画像データは、スイッチ回路130および131をそれぞれ介して字幕プレーン132およびグラフィックスプレーン133に供給される。この例では、字幕プレーン132およびグラフィックスプレーン133に供給される画像データは、PNG形式であるものとする。これらの各プレーン132、133に画像データが供給されるタイミングは、マルチメディアエンジン106により制御される。

字幕プレーン132およびグラフィックスプレーン133は、それぞれ上述した字幕プレーン11およびグラフィックスプレーン12に対応する。動画像プレーン134は、上述した動画プレーン10に対応する。字幕プレーン132、グラフィックスプレーン133および動画像プレーン134は、例えばフレームメモリからなる。

マルチメディアエンジン106は、後述するプレゼンテーションプロセッサ155に対して、動画像プレーン134、字幕プレーン13

2、ならびに、グラフィクスプレーン133を切り換える制御信号を供給する。同様に、マルチメディアエンジン106は、後述するプレゼンテーションプロセッサ157に対して、オーディオストリーム出力を制御するような制御信号を供給する。

- 5 次に、入力チャンネル(2)の系統について説明する。入力端202にMPEG2 TSで入力されたリアルタイムストリームは、PIDフィルタ110に供給され、MPEG2 TSのトランスポートパケットに格納されるPID(Packet Identification)が抽出され、当該トランスポートパケットに格納されるストリームの属性が検出される。PIDフィルタ110では、このストリーム属性に基づき、入力されたリアルタイムストリームが、トランスポートパケット毎に対応する系統に振り分けられる。

- PIDに基づき、トランスポートパケットがストアオブジェクトに属する画像データが格納されているパケットであるとされれば、当該トランスポートパケットは、バッファTBn111Aに一旦溜め込まれ、所定のタイミングで読み出されて入力端103Bが選択されたスイッチ回路103に入力され、スイッチ回路103を介してコンテンツバッファ105に格納される。

- PIDフィルタ110において、PIDに基づき、トランスポートパケットがサブピクチャデータが格納されているパケットであるとされれば、当該トランスポートパケットは、バッファTBn111BおよびバッファBn112Bに一旦溜め込まれ、所定のタイミングで読み出されて入力端107Bが選択されたスイッチ回路107に入力され、スイッチ回路107を介してグラフィクスデコーダA116に供給される。

グラフィクスデコーダA116では、供給されたトランスポートパ

ケットのヘッダ情報を除去すると共に、当該トランスポートパケットに格納されたサブピクチャデータがデコードされて字幕等を表示するための画像データとされる。この画像データは、所定のタイミングでスイッチ回路130の入力端130Bに入力され、スイッチ回路130を介して字幕プレーン132に展開される。

ネットワークを介して取得されるなどの方法により、入力端101に字幕データが入力された場合、当該字幕データは、スイッチ回路102およびスイッチ回路103を介してコンテンツバッファ105に蓄えられ、さらに、スイッチ回路107で入力端107Aが選択されることで、コンテンツバッファ105からグラフィクスデコーダA116に供給される。

PIDフィルタ110において、PIDに基づき、トランスポートパケットがグラフィクスデータが格納されているパケットであるとされれば、当該トランスポートパケットは、バッファTBn111CおよびバッファBn112Cに一旦溜め込まれ、所定のタイミングで読み出されて入力端108Bが選択されたスイッチ回路108に入力され、スイッチ回路108を介してグラフィクスデコーダB117に供給される。

グラフィクスデコーダB117では、供給されたトランスポートパケットのヘッダ情報を除去すると共に、当該トランスポートパケットに格納されたグラフィクスデータがデコードされ、グラフィクスデータとされる。この画像データは、所定のタイミングでスイッチ回路131の入力端131Bに入力され、スイッチ回路131を介してグラフィクスプレーン133に展開される。

グラフィクスデコーダA116とグラフィクスデコーダB117には機能的な違いはない。モデル上、独立して動作するグラフィクスデ



コードが2系統あることを表している。すなわち、字幕データとグラフィクスデータを独立にデコードできることを想定している。現実の実装においては、高速なグラフィクスデコードを時分割で使用し、仮想的に2系統のグラフィクスデコードが存在しているとみなす方法もある。

PIDフィルタ110において、PIDに基づき、トランスポート packets がオーディオデータが格納されている packets であるとされれば、当該トランスポート packets は、バッファTBn111DおよびバッファBn112Dに一旦溜め込まれ、所定のタイミングで読み出されてオーディオデコーダ118に供給される。このトランスポート packets に格納されるオーディオデータは、例えばMP EGに準拠した方式で圧縮符号化されている。

オーディオデコーダ118は、例えばリニアPCM(Pulse Code Modulation)オーディオデコーダ119も有する。オーディオデコーダ118は、入力されたトランスポートストリームのヘッダ情報を除去すると共に、当該トランスポート packets に格納された圧縮符号化されたオーディオデータをリニアPCMオーディオデータにデコードする。

オーディオデコーダ118から出力されたリニアPCMオーディオデータは、オーディオ用のプレゼンテーションプロセッサ157に入力され、マルチメディアエンジン106の制御に基づき所定の音響効果などが付加されて、出力端158に導出される。

PIDフィルタ110において、PIDに基づき、トランスポート packets が動画データが格納されている packets であるとされれば、当該トランスポート packets は、バッファTBn111E、バッファMBn113およびバッファEBn114に一旦溜め込まれ、所定

のタイミングで読み出されてMPEGビデオデコーダ120に供給される。このトランスポートパケットに格納される動画像データは、MPEG2方式により圧縮符号化されている。

5 MPEGビデオデコーダ120では、供給されたトランスポートパケットのヘッダ情報を除去すると共に、当該トランスポートパケットに格納された、MPEG2方式で圧縮符号化された動画像データをベースバンドの動画像データにデコードする。

10 MPEGデコーダ120から出力された動画像データは、スイッチ回路124の入力端124Aに入力されると共に、バッファ123を介してスイッチ回路124の入力端124Bに入力される。スイッチ回路124において、所定のタイミングで入力端124Aおよび124Bが選択され、出力された動画像データが動画像プレーン134に展開される。

15 PIDフィルタ110において、PIDに基づき、トランスポートパケットがシステム情報が格納されているパケットであるとされれば、当該トランスポートパケットは、バッファTBn111FおよびBs115を介してシステムデコーダ121に供給される。システムデコーダ121では、供給されたトランスポートパケットのヘッダ情報が除去され、格納されているシステム情報が取り出される。システム情報は、例えば図示されないCPUに渡される。

25 字幕プレーン132上の画像データは、上述のパレット22に対応するパレット150に供給され、256色からなるパレットに対してインデックスによる参照がなされ、RGBデータが出力されると共に、透明度データ $\alpha$ 1が抜き出される。RGBデータは上述のRGB/YCbCr変換回路29に対応するRGB/YCbCr変換回路151によりYCbCrデータに変換され、YCbCrデータおよび透明

度データ  $\alpha 1$  は、プレゼンテーションプロセッサ 155 に供給される。  
。

グラフィクスプレーン 133 上の画像データは、上述のパレット 26 に対応するパレット 152 に供給され、256 色からなるパレット  
5 に対してインデックスによる参照がなされ、RGB データが出力されると共に、透明度データ  $\alpha 2$  が抜き出される。RGB データは上述の RGB/YCbCr 変換回路 27 に対応する RGB/YCbCr 変換回路 153 により YCbCr データに変換され、YCbCr データおよび透明度データ  $\alpha 2$  は、プレゼンテーションプロセッサ 155 に供給される。  
10

動画像プレーン 134 の出力は、アップ/ダウンコンバータ 154 を介してプレゼンテーションプロセッサ 155 に供給される。

アップ/ダウンコンバータ 154 は、画像の解像度を変換する回路であって、例えば高解像度の HD (High Definition) 画像から通常の  
15 解像度を有する SD (Standard Definition) 画像への変換を行う。

プレゼンテーションプロセッサ 155 は、第 22 図を用いて説明した、字幕プレーン 11 (字幕プレーン 132) の画像データによる透明度  $\alpha 1$  と、グラフィクスプレーン 12 (グラフィクスプレーン 133) による透明度  $\alpha 2$  とを用いたアルファブレンディング処理を行う  
20 。

プレゼンテーションプロセッサ 155 では、動画像プレーン 134 の画像データに対して、字幕プレーン 132 の画像データに設定された透明度  $\alpha 1$  に基づき、字幕プレーン 132 の画像データが合成される。動画像プレーン 134 および字幕プレーン 132 が合成された画像データに対して、グラフィクスプレーン 133 の画像データに設定された透明度  $\alpha 2$  に基づき、グラフィクスプレーン 133 の画像デー  
25

タが合成される。この、グラフィクスプレーン133の画像データ、字幕プレーン132の画像データ（字幕データ）、ならびに、動画画像プレーン134の画像データが合成された画像データが出力端156に導出される。

- 5     プレゼンテーションプロセッサ155は、画像データに対してリアルタイムでエフェクト処理を行うこともできる。

字幕がサブピクチャデータとしてグラフィクスデコーダ116Aでデコードされ、字幕プレーン11に供給されるとして説明したが、字幕の表示は、この例に限定されない。例えば、字幕をテキストデータ  
10   のような文字コードで供給することもできる。文字コードは、フォントバッファ109Bに記憶されたフォントデータを参照することで、文字列を表示するためのビットマップデータに変換される。

フォントデータは、例えばディスクから再生されてストアオブジェクトとして入力端101に入力され、スイッチ回路102を介してコードバッファ104に溜め込まれ、コードバッファ104からマルチメディアエンジン106を介してフォントバッファ109Bに供給される。  
15

字幕を表示するための文字コードは、例えば、ディスクから再生されてストアオブジェクトとして入力端101から入力され、スイッチ  
20   回路102および103を介してコンテンツバッファ105に溜め込まれる。リアルタイムストリームとして入力端202から入力し、PIDフィルタ110およびバッファTBn111Aを介してスイッチ回路103に供給され、コンテンツバッファ105に溜め込まれるようにしてもよい。文字コードは、コンテンツバッファ105から読み  
25   出され、マルチメディアエンジン106に供給される。

文字コードの表示タイミングは、プログラムにより制御される。マ

マルチメディアエンジン 106 は、文字コードの表示タイミングで、表示される文字コードに基づきフォントバッファ 109 B を参照し、対応するフォントデータを選択する。例えば、文字コードが [0 x 4 1]、[0 x 4 2]、[0 x 4 3]、・・・（[0 x] は、続く数値が 16 進表記されていることを示す）となっている場合、それぞれ対応する文字「A」、「B」、「C」、・・・を表示するためのフォントデータが選択される。そして、フォントデータを基に文字の飾りやグリフ形状を変化させ、プログラムで指定された大きさのビットマップデータを生成する（レンダリングと称する）。

- 10 生成されたビットマップデータは、スイッチ回路 130 を介して字幕プレーン 132 に供給される。これは、字幕表示は、動画像プレーン 134 上の動画と同期を取る必要があるためである。

文字コードのレンダリングは、マルチメディアエンジン 106 やシステムの CPU で行うのに限らず、専用のハードウェアを設けて行ってもよい。文字コードが参照するフォントデータは、ディスクから再生されたものに限らず、例えばネットワークを介して取得するようにしてもよいし、プレーヤのハードウェアが有する ROM (Read Only Memory) などに予め記憶させておいてもよい。ユーザによりフォントデータの種類を選択させるようにすることも可能である。

- 20 字幕を文字コードで供給することで、予め画像データで供給される字幕データに比べ、字幕表示に要するデータ量が非常に少なくて済むメリットがある。

プレーヤデコーダ 100 の各部がハードウェアで構成されるように説明したが、これはこの例に限られない。例えば、プレーヤデコーダ 100 をソフトウェア上の処理として実現することも可能である。この場合、プレーヤデコーダ 100 をコンピュータ装置上で動作させる

ことができる。プレーヤデコーダ100をハードウェアおよびソフトウェアが混合された構成で実現することもできる。例えば、オーディオデコーダ118やMP E Gビデオデコーダ120をハードウェアで構成し、その他をソフトウェアで構成することが考えられる。

- 5 プレーヤデコーダ100をソフトウェアのみ、または、ハードウェアおよびソフトウェアの混合により構成し、コンピュータ装置で実行させるためのプログラムは、例えばCD-ROM (Compact Disc-Read Only Memory) といった記録媒体に記録されて提供される。このCD-ROMをコンピュータ装置のCD-ROMドライブに装填し、CD-ROMに記録されたプログラムを所定のコンピュータ装置にインストールすることで、上述の処理をコンピュータ装置上で実行可能な状態とすることができる。コンピュータ装置の構成は、極めて周知であるため、説明は省略する。

15 2-10. ボタンについて

- この発明の実施の一形態によるユーザインターフェイスについて、より詳細に説明する。第74図は、グラフィクスプレーン12上に表示されるボタン表示の一例の状態遷移図である。ボタン表示の状態は、画面にボタンが表示されるボタン表示状態と、画面にボタンが表示されていないボタン非表示状態とに大別される。ボタン非表示状態からボタン表示が開始されボタン表示状態に遷移する。ボタン表示状態からボタン表示が終了され、ボタン非表示状態に遷移する。ボタン表示状態は、さらに、通常状態、選択状態および実行状態の3種類の状態を有する。ボタン表示は、これら3種類の状態間をそれぞれ互いに遷移することができる。遷移方向を一方向に限定することも可能である。

上述の第25図を用いて、より具体的に説明する。先ず、ディスク挿入時や、ユーザがリモコンのメニューキーを押したときに、メニュー画面60が表示される。メニュー画面60の表示に伴い、ボタン62A、62B、62C、63D、64および65が非表示状態から表示状態へと遷移する。ボタン表示開始時は、ボタン62A、62B、62C、63D、64および65のうち一つが予め選択状態にされていることが多い。例えばボタン62Aが予め選択状態とされ、他のボタンが通常状態とされているものとする。

この状態で、ユーザがリモコンの例えば矢印キーを操作すると、通常状態のボタンの一つ（例えばボタン62B）が選択状態に遷移すると共に、選択状態だったボタン62Aが通常状態へと遷移する。このようにして、ユーザ操作によってカーソルが移動していく。ユーザがリモコンの決定キーを操作すると、そのときに選択状態にあったボタン62Bが選択状態から実行状態へと遷移し、当該ボタン62Bに予め割り当てられているプレーヤ動作が実行される。

上述したように、プレーヤ動作は、独自コマンドによるプログラミング言語やECMAScriptなどのスクリプト言語により記述され、ディスク上に記録され用意されている。このプレーヤ動作に関する記述は、独立したファイルとしてディスク上に記録してもよいし、後述するグラフィックオブジェクトのように、クリップAVストリームファイルに多重化することも可能である。プレーヤ動作に関する記述をネットワークを介してプレーヤ内部のメモリやストレージ装置にダウンロードすることも考えられる。

このようなメニュー画面を構成するボタンなどの画像データおよび当該画像データに付随する制御情報のデータ構造について説明する。ディスクに収録されるコンテンツ本体をなす動画以外に表示される映

像要素として、字幕およびグラフィクス（静止画）を考える。画面に表示される字幕やグラフィクスといった要素をオブジェクトとして考え、オブジェクトの種類を、第75図に示されるように、字幕、同期型グラフィクスおよび非同期型グラフィクスの3種類に分類する。

- 5     字幕とは、例えば映画の字幕のように、動画と同期して表示／非表示の切り換えが行われ、リモコンなどによるユーザ入力とは無関係な映像要素を指す。グラフィクスとは、メニュー画面上のボタンなどのように、ユーザ入力を受け付けることができる映像要素とする。グラフィクスは、同期型グラフィクスおよび非同期型グラフィクスの2種類に分類される。同期型グラフィクスとは、例えば本編再生中のある
- 10    タイミングに表示される分岐選択画面のように、動画と同期して表示される映像要素とする。非同期型グラフィクスとは、ディスク挿入時に最初に表示されるメニュー画面や、ユーザ入力に応じて表示される画面など、本編の再生とは非同期に表示される映像要素を指す。Java VM上で動作するJavaアプリケーションにより表示される映像要素や、ブラウザソフトウェア上でHTMLファイルの記述に基づ
- 15    き表示される映像要素は、非同期グラフィクスである。

- 各映像要素と動画プレーン10に表示される本編映像との関係からい
- 20    えば、字幕および同期型グラフィクスは、本編映像に同期して表示されるため、共に同期型である。一方、非同期型グラフィクスは、上述のように本編映像と非同期で表示されるので、その名の示す通り、非同期型である。

- 字幕とグラフィクスとは、表示先のプレーンの違いから分類することができる。字幕は、字幕プレーン11に表示される。同期型および
- 25    非同期型グラフィクスは、グラフィクスプレーン12に表示される。

字幕および同期型グラフィクスは、本編の動画再生中に表示すると



いう共通の性質を持つことから、共通のデータ構造とすることが好ましい。以下では、これら共通のデータ構造とされる字幕および同期型グラフィックスを、グラフィックスオブジェクトと称する。グラフィックスオブジェクトは、常に動画再生と同期して表示されるため、動画と同一のストリームに多重化しておく、扱いが容易になる。

第76図A、第76図Bおよび第76図Cは、この発明の実施形態におけるグラフィックスオブジェクト200のデータ構造の例を示す。グラフィックスオブジェクト200は、第76図Aに一例が示されるように、グラフィックスオブジェクトヘッダ201、表示制御命令テーブル202およびPNGデータ領域203、音声データ領域204の4つの領域からなる。

以下の例では、グラフィックスオブジェクト200で扱われる画像データの形式をPNG形式とし、画像データをPNG画像データとしているが、JPEG形式なビットマップデータ、ランレングス圧縮された画像データ、圧縮符号化がなされていないビットマップデータなど、他の形式の画像データをグラフィックスオブジェクト200で扱うことも可能である。便宜上、画像データをPNG画像、PNG画像データなどと表現する。

第76図A、第76図Bおよび第76図Cにおいて、グラフィックスオブジェクトヘッダ201は、当該グラフィックスオブジェクト200の属性を表す情報が格納される。グラフィックスオブジェクト200の属性は、例えば、当該グラフィックスオブジェクト200のデータサイズ、当該グラフィックスオブジェクト200が有するPNG画像数、当該グラフィックスオブジェクト200が有するPNG画像データが共通に使用するパレットデータ、ならびに、当該グラフィックスオブジェクト200を識別する識別情報からなる。識別情報は、例えばグラフィ

クスオブジェクト 200 毎に一意に割り当てられた番号である。グラフィクスオブジェクトヘッダ 201 に、さらに他の情報を格納してもよい。

表示制御命令テーブル 202 は、当該グラフィクスオブジェクト 200 が有する PNG 画像の表示位置、表示開始時刻および表示終了時刻といった、PNG 画像の表示を制御するための情報が格納されるテーブルである。

PNG データ領域 203 は、PNG 形式で圧縮符号化された画像データ（以下、PNG データと称する）が格納される。PNG データ領域 203 には、複数の PNG データ 203A、203B、・・・、203n を格納することができる。PNG データ領域 203 に格納される PNG データ 203 の数は、グラフィクスオブジェクトヘッダ 201 に記述されている。

PNG データ領域 203 に格納される複数の PNG データ 203A、203B、・・・、203n は、アニメーションを構成する複数枚の静止画の組や、上述したボタン表示の 3 状態を表す画像といった、相互に関連の強い画像であることを想定している。これらを、一つのグラフィクスオブジェクト 200 に纏めることで、PNG 画像の扱いを容易にすることができる。

音声データ領域 204 については、後述する。

グラフィクスオブジェクト 200 は、表示が可能になる時刻を示す時刻情報を有する。リアルタイムストリームが MPEG2 TS で伝送されるこの例では、この時刻情報として、MPEG2 (Moving Pictures Experts Group 2) により定義された、pts (Presentation Time Stamp) を用いる。pts は、再生出力の時刻管理情報で、90 kHz のクロックで計測した値を 33 ビット長で表す。MPEG システム

の基準復号器内部のS T C (System Time Clock)がp t sに一致したときに、対象となるアクセスユニットを再生出力する。一つのグラフィクスオブジェクト200は、p t sで表される時刻から表示が可能になり、その時刻以降に、表示制御命令により実際に表示のO N / O F Fが行われる。表示制御命令により表示の管理を行うため、表示を停止した後に再度、同じグラフィクスオブジェクト200を表示させるような制御が可能である。

第76図Bは、字幕を格納するグラフィクスオブジェクト200の例を示す。P N Gデータ領域203に、字幕として表示する画像がP N Gデータ(1)203A-1として格納される。通常、字幕であれば、一つのグラフィクスオブジェクト200に含まれるデータは、P N Gデータ(1)203A-1一つで十分である。表示制御命令テーブル202に対して、P N Gデータ(1)A-1の表示属性(透明度、表示色など)を変化させる命令を加えることで、P N Gデータ(1)A-1自体を変更せずに、P N Gデータ(1)A-1の表示を変化させることができる。

例えば、字幕に対して、フェードイン/フェードアウトのような、画像の内容が変化しない特殊効果を施すときは、表示制御命令テーブル202に対してP N Gデータ(1)A-1の透明度を変化させる表示制御命令を加えることで、実現できる。フェードイン/フェードアウトを行う際に、P N Gデータ(1)A-1自体を変更する必要が無い。同様に、表示制御命令テーブル202に対して、P N Gデータ(1)A-1が参照するパレットデータを変更する表示制御命令を加えることで、P N Gデータ(1)A-1自体を変更せずに、表示色だけを変化させることができる。

字幕に対して、アニメーションなどのような、画像そのものが変化

5 するような効果を施す場合には、第76図Bに点線で示されるように、アニメーションの各動作に対応する複数のPNGデータ(2)B-1、PNGデータ(3)C-1、PNGデータ(4)D-1、...を、一つのグラフィクスオブジェクト200に格納すればよい。これに限らず、日本語字幕、英語字幕など、言語が異なる字幕を表示するPNGデータをPNGデータ(1)A-1、PNGデータ(2)B-1、...として、一つのグラフィクスオブジェクト200に格納することもできる。

10 第76図Cは、ボタンを構成するグラフィクスオブジェクト200の例を示す。ボタンは、上述したように、通常状態、選択状態および実行状態の3種類の状態を有し、これら3種類の状態のそれぞれでボタン表示を異ならせることができる。3種類の状態でボタン表示を異ならせる場合、3枚のボタン画像データを用意しておく必要がある。これら3枚のボタン画像データを、ひとまとまりのグラフィクスオブジェクト200として扱う。グラフィクスオブジェクト200のPNGデータ領域203に、通常状態、選択状態および実行状態それぞれのボタン表示に用いるPNGデータ203A-2、203B-2および203C-2を格納する。

20 第76図Bに実線で示される通常の子幕のように、グラフィクスオブジェクト200内に一つだけ字幕表示のPNGデータ203A-1が存在する場合には、当該グラフィクスオブジェクト200内の表示制御命令テーブル202に格納される表示制御命令は、そのPNGデータ203A-1に対するものとなる。第76図Cに一例が示されるような、グラフィクスオブジェクト200内に、複数のPNGデータ203A-2、203B-2および203C-2が存在する場合、表示制御命令テーブル202に格納される表示制御命令が、複数のPN

Gデータ203A-2、203B-2および203C-2のうちどのデータに対する表示制御命令であるかを特定する必要がある。

例えば、第76図Cに例示される、ボタンのグラフィクスオブジェクト200においては、このボタンの表示開始時の初期状態が選択状態と定められている場合は、最初に表示するボタン画像は、PNGデータ領域203の先頭に配置される、通常状態時のPNGデータ203A-2ではなく、選択状態時のPNGデータ203B-2としなければならない。この発明の実施の第1の形態では、このような表示制御は、グラフィクスオブジェクト200の外で行うようにしている。

- 10     例えば、各ボタンの初期状態や、表示開始および表示停止、ボタンが実行状態に遷移された際に実行されるプログラムなどは、グラフィクスオブジェクト200外部のスクリプトプログラム、例えば上述のECMASクリプトやJavaScriptによって指定する方法が考えられる。また、ボタンの表示中における、ボタン表示を行うPNG
- 15     Gデータの変更は、例えばユーザがリモコンの矢印キーを操作し、カーソルを移動させた際に発生する。この場合には、プレーヤがユーザ入力に応じて各ボタンのPNGデータを入れ替えることになる。

- この発明の実施の一形態では、グラフィクスオブジェクト200は、MPEG2に規定されるパケットに分割され、クリップAVストリームに多重化され、クリップAVストリームファイルとしてディスクに記録される。第77図に一例が示されるように、グラフィクスオブジェクト200は、分割され、MPEG2に規定されるPES (Packetized Elementary Stream)パケット210、210、・・・に格納される。このとき、グラフィクスオブジェクトヘッダ201および表示
- 20     制御命令テーブル202、ならびに、各PNGデータ203A、203B、・・・、203nそれぞれの先頭が、PESパケット210、
- 25

210、・・・のペイロードの先頭に合致するように、データが配置される。こうすることで、再生時に、グラフィクスオブジェクト200内の各データの検索が容易となる。

5 こうして分割されてPESパケット210、210、・・・に格納されたグラフィクスオブジェクト200は、さらに、データサイズが188バイトに固定されたTSパケットに分割され（図示しない）、クリップAVストリームなどの動画データや音声データによるストリームに多重化される。

第78図は、グラフィクスオブジェクト200のデコードを行うグラフィクスオブジェクトデコーダモデル240の一例の構成を示す機能ブロック図である。このグラフィクスオブジェクトデコーダモデル240は、第73図A、第73図Bおよび第73図Cを用いて説明したプレーヤデコーダ100におけるマルチメディアエンジン106およびグラフィクスデコーダA116を中心に構成される。グラフィクスデコーダB117の機能は、グラフィクスデコーダA116と同一であるので、ここでの説明では、グラフィクスデコーダA116を想定することにする。第78図において、第73図A、第73図Bおよび第73図Cと共通する部分には同一の符号を付して詳細な説明を省略する。この第78図は、より機能面に注目して表現されており、上述の第73図A、第73図Bおよび第73図Cとは異なる表現がなされている場合があるので、必要に応じて第73図A、第73図Bおよび第73図Cの構成と対応付けながら説明する。

25 端子202からMPEG TSとして入力されたクリップAVストリームがPIDフィルタ110に供給される。PIDフィルタ110は、MPEG2 TS（トランスポートストリーム）に対するデマルチプレクサとして機能し、供給されたMPEG TSのPIDに基づ

き、動画像データ、オーディオデータおよびグラフィクスオブジェクト 200 をそれぞれ抜き出す。動画像データは、ビデオバッファとしてのバッファ T B n 1 1 1 E に供給される。オーディオデータは、オーディオバッファとしてのバッファ 1 1 1 D に供給される。グラフィクスオブジェクト 200 は、グラフィクスオブジェクト（第 78 図では「GOBJ」と表記）の入力バッファであるバッファ T B n 1 1 1 B に供給される。

グラフィクスオブジェクト 200 は、バッファ T B n 1 1 1 B から読み出され、GOBJ パーサ 224 に供給される。GOBJ パーサ 224 は、例えば第 73 図におけるグラフィクスデコーダ A 1 1 6 の機能の一つである。GOBJ パーサ 224 では、供給されたグラフィクスオブジェクト 200 のグラフィクスオブジェクトヘッダ 201 を読み取り、パレットデータを抽出すると共に、表示制御命令テーブル 202、PNG データ領域 203 および音声データ領域 204 を分離する。パレットデータおよび表示制御命令テーブル 202 は、コマンドプロセッサ/グラフィックレンダラ 225 に供給される。PNG データ領域 203 の PNG データ 203 A、203 B、・・・は、PNG デコーダバッファ 226 に一時的に格納される。PNG デコーダバッファ 226 は、例えば第 73 図 A、第 73 図 B および第 73 図 C におけるバッファ B n 1 1 2 B に対応する。

音声データ領域 204 の音声データ 204 A、204 B、・・・、204 n はコマンドプロセッサ/グラフィックレンダラ 225 に供給され、図示されないバッファにそれぞれ溜め込まれる。

PNG デコーダバッファ 226 に格納された PNG データ 203 は、グラフィクスデコーダ A 1 1 6 の機能の一つである PNG デコーダ 227 によりデコードされ、ビットマップデータとされる。このビッ

トマップデータは、オブジェクトバッファ 228 に蓄積される。オブジェクトバッファ 228 は、第 73 図 A、第 73 図 B および第 73 図 C においては、デコーダ 116 が内部で持つバッファメモリに対応する。ソフトウェアによるデコードにおいては、第 73 図 A、第 73 図 B および第 73 図 C におけるグラフィクスバッファ 109 D に対応する。

コマンドプロセッサ／グラフィックレンダラ 225 は、GOBJ パーサ 224 から供給された表示制御命令テーブル 202 に記述される表示制御命令に従い、オブジェクトバッファ 228 に蓄積されているビットマップデータを読み出して指定された時刻にプレーンバッファ 229 に転送する。プレーンバッファ 229 は、例えば第 73 図 A、第 73 図 B および第 73 図 C における字幕プレーン 132 およびグラフィクスプレーン 133 に対応させることができる。例えば、字幕と、字幕以外のグラフィクスオブジェクトとに対してそれぞれプレーンバッファ 229 A、229 B (図示しない) を設けることができる。これに限らず、字幕プレーン 132 およびグラフィクスプレーン 133 を、プレーンバッファ 229 上の異なる領域とすることもできる。

コマンドプロセッサ／グラフィックレンダラ 225 は、GOBJ パーサ 224 から供給されたパレットデータを、第 73 図 A、第 73 図 B および第 73 図 C のパレット 150 に対応する共通パレットテーブル 230 に供給する。コマンドプロセッサ／グラフィックレンダラ 225 は、第 73 図 A、第 73 図 B および第 73 図 C のマルチメディアエンジン 106 の一部の機能およびグラフィクスデコーダ A116 の一部の機能を併せ持つ。

コマンドプロセッサ／グラフィックレンダラ 225 は、GOBJ パーサ 224 から供給された表示制御命令テーブル 202 に記述される



表示制御命令に従い、バッファから音声データを読み出し、出力する。  
グラフィクスオブジェクト 200 に格納された音声データ 204 A、  
204 B、・・・、204 n が圧縮符号化されている場合には、コ  
マンドプロセッサ／グラフィックレンダラ 225 において復号化され  
5 て出力される。

コマンドプロセッサ／グラフィックレンダラ 225 から出力された  
音声データは、オーディオミキサ 231 に供給され、プレゼンテーシ  
ョンプロセッサ 157 に対して出力される。オーディオミキサ 231  
に他の音声データも入力されている場合には、これらの音声データが  
10 所定の割合で混合されて出力される。

グラフィクスオブジェクト 200 がボタンを構成するものである場  
合、ボタンの 3 種類の状態にそれぞれ対応した PNG データ 203 A  
、203 B および 203 C がグラフィクスオブジェクト 200 に格納  
される。PNG データ 203 A、203 B および 203 C は、PNG  
15 デコーダ 227 でデコードされ、オブジェクトバッファ 228 にそれ  
ぞれ蓄積される。

ユーザのリモコンなどによる入力が、コマンドプロセッサ／グラフ  
ィックレンダラ 225 により受け取られる。コマンドプロセッサ／グ  
ラフィックレンダラ 225 は、このユーザ入力に応じて、オブジェク  
20 トバッファ 228 から該当するビットマップを読み出して、プレーン  
バッファ 229 に転送する。例えば、ユーザ入力があるボタンに対し  
て選択状態から実行状態に状態遷移させるものであれば、オブジェク  
トバッファ 228 から、実行状態のボタン画像に対応するビットマッ  
プデータが選択的に読み出されて、プレーンバッファ 229 に転送さ  
25 れる。

コマンドプロセッサ／グラフィックレンダラ 225 は、オブジェク

トバッファ 228 から読み出したビットマップデータに対して、表示制御命令に従い部分切り出しなどの特殊効果処理を施すこともできる。

この実施の一形態では、PNGデータは、1画素のサンプリング深さが8ビットとされているため、プレーンバッファ 229 は、1画素当たり8ビットのデータが並んでいることになる。プレーンバッファ 229 は、ディスプレイ装置などに実際に画像を表示させる処理を行う表示系の走査周期毎に読み出される。プレーンバッファ 229 から読み出されたビットマップデータは、例えば第73図A、第73図Bおよび第73図Cのパレット150に対応できる共通パレットテーブル 230 に供給され、パレットインデックス値から実際のRGB (4:4:4) の色情報に変換されると共に、透明度データ  $\alpha 1$ 、 $\alpha 2$  が抜き出される。RGB (4:4:4) の色情報は、図示されない変換回路でYCbCr (4:4:4) の色情報に変換され、透明度データ  $\alpha 1$ 、 $\alpha 2$  と共に、第73図A、第73図Bおよび第73図Cのプレゼンテーションプロセッサ 155 に供給される。

例えばフェードイン/フェードアウトなどの、パレットや透明度を変える処理が必要な特殊効果は、コマンドプロセッサ/グラフィックレンダラ 225 が表示制御命令に従い共通パレットテーブル 230 のデータを変化させることで実現される。字幕と、字幕以外のグラフィクスオブジェクト 200 に対して、それぞれ共通パレットテーブル 230 A、230 B (図示しない) を設けてもよい。

第79図A、第79図B、第79図Cおよび第79図Dは、グラフィクスオブジェクト入力バッファ (バッファ TBn111B、以下GOBJ入力バッファ)、PNGデコーダ 227、オブジェクトバッファ 228 およびプレーンバッファ 229 における蓄積データ量の遷移

の例を概略的に示す。PNGデコーダ227は、PNGデコーダ227がPNGデータのデコード時に用いるバッファにおける蓄積データ量が示されている。

第79図A、第79図B、第79図Cおよび第79図Dの例では、  
5 3つのグラフィクスオブジェクトGOBJ#1、GOBJ#2およびGOBJ#3によるデータ量の遷移が時間の経過に従い示される。グラフィクスオブジェクトのデコード開始時刻は、MP EG 2システムにおけるdts (Decoding Time Stamp)で表される。オブジェクトの有効期間は、開始時刻がptsで表され、グラフィクスオブジェクト  
10 ヘッダ201内に記述される時刻presentation\_endで終了される。この有効期間内に、表示制御命令によって画像の表示開始および終了が指示される。

第79図Dを参照し、GOBJ入力バッファにグラフィクスオブジェクトGOBJ#1のPNGデータが入力され、時刻dts of GOBJ#1  
15 に、このPNGデータのデコードが開始される。第79図Cを参照し、PNGデータがGOBJ入力バッファからPNGデコーダ227に転送され、PNGデータがビットマップデータにデコードされる。実際には、PNGデータは、GOBJ入力バッファから一旦PNGデコーダバッファ226に溜め込まれ、PNGデコーダ227は、PNG  
20 デコーダバッファ226に溜め込まれたデータを用いてデコード処理を行う。

PNGデコーダ227は、デコード速度に上限があるため、GOBJ入力バッファからPNGデコーダバッファ226に対して、PNGデコーダ227のデコード速度を超えないような転送速度でデータが  
25 供給される。そのため、PNGデコーダバッファ226に対して、PNGデコーダ227への転送時間を0とした概念モデル上の場合の垂

直線に対し、ある傾きAに対応するデータ転送レートで以てPNGデータが入力される。

PNGデータのPNGデコーダ227への入力が完全に終了していても、PNGデータのデコードを開始することが可能である。この第79図A、第79図B、第79図Cおよび第79図Dの例では、GOBJ入力バッファに格納されたオブジェクトGOBJ#1が全てPNGデコーダ227に転送された後、次のオブジェクトGOBJ#2のPNGデータの、GOBJバッファへの入力が始まっている。

オブジェクトGOBJ#2およびオブジェクトGOBJ#3についても、ある傾きBおよびCにそれぞれ対応する転送レートで以て、PNGデコーダバッファ226に対してPNGデータが入力される。傾きBは、実際には、複数の区間でそれぞれ異なる傾きとなっている。

時刻pts of GOBJ#1によりオブジェクトGOBJ#1の有効期間が開始されると、デコードされPNGデコーダバッファに蓄積された、オブジェクトGOBJ#1のビットマップデータがオブジェクトバッファ228に転送される（第79図B）。オブジェクトバッファ228に転送されたオブジェクトGOBJ#1は、このオブジェクトGOBJ#1に対して時刻presentation end of GOBJ#1（GOBJ#1の提示終了）で示される時刻まで、有効期間が持続される。

オブジェクトGOBJ#1の有効期間内に、命令Display ON cmd. of GOBJ#1（GOBJ#1の表示開始指示）によりオブジェクトGOBJ#1の表示命令が出されたら、オブジェクトバッファ228に蓄積されたオブジェクトGOBJ#1のビットマップデータがプレーンバッファ229に転送され、表示される（第79図A）。詳細は後述するが、ビットマップデータをプレーンバッファ229へ転送する際の転送レートは、バス幅などの影響により上限を有する。そのため、プ

レーンバッファ 229 に対するビットマップデータの書き込みは、例えばある傾き D に対応する転送レートで以て行われる。

他のオブジェクト GOBJ # 2 やオブジェクト GOBJ # 3 についても同様に、ある傾き E、F、G に対応する転送レートで以てビットマップデータが転送され、プレーンバッファ 229 に書き込まれる。

表示は、オブジェクト GOBJ # 1 に対して表示停止を命令するコマンド Display OFF cmd. of GOBJ#1 (GOBJ # 1 の表示終了指示) が出されるまで、持続される。コマンド Display OFF cmd. of GOBJ#1 が出されると、プレーンバッファ 229 に格納された当該オブジェクト GOBJ # 1 のビットマップデータが破棄され、表示が停止される。

GOBJ バッファには、オブジェクト GOBJ # 2 および GOBJ # 3 の PNG データも、順次、入力される。そして、オブジェクト GOBJ # 1 の場合と同様に、時刻 dts of GOBJ#2、時刻 dts of GOBJ#3 からそれぞれデコード開始が指示され、PNG データが PNG デコーダ 227 に供給され、PNG デコーダバッファを用いてビットマップデータへのデコードが行われる。そして、オブジェクト GOBJ # 2 に対して時刻 pts of GOBJ#2 により有効期間開始が指示され、第 79 図では省略されているが、コマンド Display ON cmd. of GOBJ#2 によりオブジェクト GOBJ # 2 の表示命令が出され、オブジェクトバッファ 228 からプレーンバッファ 229 にビットマップデータが転送され、コマンド Display OFF cmd. of GOBJ#2 が出されるまで、オブジェクト GOBJ # 2 が表示される。

この第 79 図 A、第 79 図 B、第 79 図 C および第 79 図 D の例では、オブジェクト GOBJ # 2 は、一旦、図示されないコマンド Display OFF cmd. of GOBJ#2 により表示が停止された後、再びコマンド Dis

play ON cmd. of GOBJ#2による表示が再開されている。オブジェクトバッファ228に格納されたオブジェクトGOBJ#2のビットマップデータは、オブジェクトGOBJ#2に対して有効期間終了時刻presentation end of GOBJ#1を指定するまで保持されるため、コマンド

5 Display ON cmd. of GOBJ#2を与えることにより、繰り返しオブジェクトGOBJ#2を表示させることができる。

オブジェクトGOBJ#3は、オブジェクトGOBJ#2の有効期間中に重複して、有効期間が指定されている。この場合、オブジェクトバッファ228は、空き容量に応じて、複数のビットマップデータを互いに異なる領域に格納する。例えば、オブジェクトバッファ228からオブジェクトGOBJ#2のビットマップデータをプレーンバッファ229に転送して表示中に、オブジェクトバッファ228の異なる領域からオブジェクトGOBJ#3のビットマップデータをプレーンバッファ229に転送することで、2枚のビットマップデータを

10 同時に表示させることができる。

## 2-11. グラフィックスの転送速度について

このグラフィックスオブジェクトデコーダモデル240（以下、デコーダモデル240と略称する）をプレーヤに実装する場合について考

20 える。同一ディスクを異なるプレーヤで再生した際の、プレーヤ同士の再生互換性を保つためには、デコーダモデル240に対して所定の制限を設けることが必要となる場合がある。例えば、デコーダモデル240のグラフィックス処理能力には上限があり、その能力を超えるグラフィックスデータが入力された場合には、グラフィックスデータのデ

25 コードを完全に行うことができなくなり、正常な表示を行うことができなくなる。

プレーヤ側では、プレーヤが最小限備える必要のあるグラフィクス処理能力を規格により定める。一方、ディスク側では、ディスクに記録するコンテンツを制作する側において、規格により定められたプレーヤの能力の範囲内で処理可能なグラフィクスを用意する。このよう  
5 にしてプレーヤ側とディスク制作側とでグラフィクス処理能力の整合性を図ることで、再生互換性を保つことができる。

この発明の実施の一形態では、上述の第78図における、GOBJパーサ224からPNGデコーダバッファ226へのデータ転送速度R(1)と、コマンドプロセッサ225からプレーンバッファ229  
10 へのデータ転送速度R(2)とを規定する。

データ転送速度R(1)は、PNGデコーダバッファ226に入力されるデータの、単位時間当たりのデータ転送量を規定する。すなわち、上述の第79図Cに示される傾きA、BおよびCがデータ転送速度R(1)に対応する。これは、PNGデコーダバッファ226の後  
15 に接続されるPNGデコーダ227の、圧縮符号化されたグラフィクスデータを単位時間当たりにデコードできる量を示すデコード能力を規定する。したがって、データ転送速度R(1)を制限することで、入力された圧縮符号化グラフィクスデータに対してデコードが間に合わなくなり、表示などが破綻状態になる状況が防止される。

20 データ転送速度R(2)は、画像の更新速度を規定する。プレーンバッファ229は、実際に表示装置に表示される画面に対応している。プレーンバッファ229へのデータの書き込み速度によって、ユーザが見るグラフィクスの更新速度が決まる。データ転送速度R(2)の単位は、[バイト/秒]であって、プレーン全体すなわち全画面を  
25 更新する際の最小更新時間間隔を規定する。上述の第79図Aに示される傾きD、E、FおよびGがデータ転送速度R(2)に対応する。

プレーンの一部だけを更新する場合には、更新される画像データが少ないためデータ転送速度  $R(2)$  で規定される最小更新時間間隔よりも短い周期で更新される。更新間隔は、更新される画像データのデータ量に反比例するとは限らず、当該画像データのプレーン上の配置により大きく影響される。

プレーンの更新速度について、第 80 図を用いてより具体的に説明する。オブジェクトバッファ 228 には、2 個のグラフィクスオブジェクト 460 および 461 が格納されているものとする。これら 2 個のグラフィクスオブジェクト 460 および 461 を、同時にプレーンバッファ 229 に書き込み表示させることを考える。

グラフィクスオブジェクト 460 および 461 は、オブジェクトバッファ 228 から読み出され、コマンドプロセッサ/グラフィックレンダラ 225 に供給される。このコマンドプロセッサ/グラフィックレンダラ 225 からの出力が上述のデータ転送速度  $R(2)$  により制限され、画面上の更新速度（更新間隔）が制限されることになる。

しかしながら、書き換えデータ量が同一のオブジェクトであっても、画面上の更新速度は、オブジェクトのプレーン上の位置、さらにオブジェクトの変形や移動などによって変化してしまい、見積もることが難しい。第 80 図の例では、更新速度は、グラフィクスオブジェクト 460 および 461 の合計のデータ量ではなく、グラフィクスオブジェクト 460 および 461 を、プレーン上にどのように配置するかによって決まる。

そこで、プレーンバッファへの書き換えデータ量について、ウィンドウと称する矩形の更新領域を定義した。これにより、以下に説明するように、最小更新時間間隔を見積もることが可能となり、実装の実現性と再生互換性を高めることができるようになる。定義される領域



が矩形であるため、グラフィクス処理を行う一般的なグラフィクスプロセッサに対して容易に適用可能で実現性が高い考え方である。以下、このウィンドウの定義に基づくモデルをウィンドウモデルと称する。

- 5     例えば、第80図においては、プレーン上に配置されるグラフィクスオブジェクト460および461を全て含む矩形領域462で以て、プレーンの更新がなされる。コマンドプロセッサ／グラフィックレンダラ225において、グラフィクスオブジェクト460および461の配置情報に基づき矩形領域462の画像データが形成される。この矩形領域462の画像データが例えば転送バスを介してプレーンバッファ229に供給される。プレーンバッファ229では、例えば表示位置の指定に基づき、矩形領域462に対応する領域のデータが、新たに供給された矩形領域462のデータと置き換えられる。

- 15     コマンドプロセッサ／グラフィックレンダラ225から出力される画像データは、ビットマップデータであるので、画像の内容によらず、画像の面積に応じたデータ量を有することになる。第80図の例の場合、2個のグラフィクスオブジェクト460および461を含む矩形領域462の画像データ量は、例えば幅(Width)および高さ(Height)をそれぞれ画素数で表すとき〔幅(Width)×高さ(Height)〕バイトで表すことができる。この矩形領域462をウィンドウと呼ぶことにする。ウィンドウは、1または複数個のグラフィクスオブジェクトを完全に含むと共に、面積が最小となるようにすると、転送データ量が最小になり好ましい。

- 25     プレーンバッファ229へのデータ転送速度が速度R(2)〔バイト／秒〕とされていることから、これら2個のグラフィクスオブジェクト460および461は、{速度R(2)／(幅×高さ)}秒の時

間で更新できることが分かる。ある幅および高さのウィンドウがプレーンバッファ 229 に転送されてから少なくとも  $\{ \text{速度} R(2) / (\text{幅} \times \text{高さ}) \}$  秒後には、次のグラフィクスオブジェクトの描画が可能となる。ディスク制作側では、次のグラフィクスオブジェクトの描画を、少なくともこの時間以上の間隔を空けて行うようにプログラムすること、どのプレーヤでも同一のグラフィクス表示がなされ、再生互換性を保つことができる。

以上のように、同時に表示制御される複数オブジェクトを囲む矩形領域をウィンドウと定義し、ウィンドウのデータ量を転送速度  $R(2)$  で割ることにより、ウィンドウの最短更新間隔を見積もることができる。

オブジェクトを消去する際にも、プレーンへの書き込みが必要になる。上述したウィンドウモデルにおいては、ウィンドウ全体を透明色で書き換えればよい。そのときの時間も、ウィンドウの最短更新間隔と同じである。

オブジェクトが変形する、あるいはオブジェクトがプレーン上を移動するグラフィクスを制作したい場合でも、上述したウィンドウモデルは、動作速度を見積もることができる。例えば、第 81 図 A のようにオブジェクトが時間と共に変形する場合は、第 81 図 B のように、時間により変化するオブジェクトを全て含む矩形領域  $(x_{\min}, y_{\min})$  ,  $(x_{\max}, y_{\max})$  をウィンドウと定義すればよい。例えば、第 82 図 A の例のようにオブジェクトが時間と共に移動する場合は、第 82 図 B のように、オブジェクトが移動する軌跡を全て含む矩形領域  $(x_{\min}, y_{\min})$  ,  $(x_{\max}, y_{\max})$  をウィンドウと定義すればよい。

ウィンドウは、プレーンに複数個（例えば 2 個）作ることが許される。このとき、同一プレーン上の複数個のウィンドウは、互いに重な

り合わないものとする。

この発明の実施の一形態では、以上のようなウィンドウモデルを定義することによって、表示するオブジェクトの数、形状、大きさ、時間経過による変形、表示位置など、多くのパラメータで決まるために  
5 算出が難しかったグラフィックスの表示速度（最小更新間隔）を簡単に導出することができるようになる。そのため、ディスク制作側がグラフィックスの動作速度を予め見積もることができるようになり、プレーヤでの動作の互換性を高めることが可能となる。

後述する動きのある字幕表示についても、上述のようにしてデータ  
10 転送速度R（2）を見積もることで、再生互換性を保つようにアニメーション速度を決めることができる。

## 2-12. グラフィックスオブジェクトについて

グラフィックスオブジェクト200の構造について、より詳細に説明  
15 する。第83図は、グラフィックスオブジェクト200の一例の構造を表すシンタクスを示す。第76図Aなどにおけるグラフィックスオブジェクトヘッダ201がブロックGraphicsObjectHeader()、表示命令制御テーブル202がブロックGOBJCommandTable()、PNGデータ領域  
203がブロックPNGImageRegion()、音声データ領域204がブロッ  
20 クSoundDataRegion()に相当する。

ブロックGraphicsObjectHeader()の先頭にフィールドlengthが配置される。フィールドlengthは、データ長が8ビットの0以上の整数で、フィールドlengthの直後からブロックGraphicsObjectHeader()の終  
わりまでの長さをバイトで示す。フィールドpresentation\_end\_time\_  
25 stampは、データ長が33ビットの0以上の整数で、このグラフィックスオブジェクト200の有効期間終了時刻を表す。グラフィックスオブ

- ジェットの有効期間は、PESパケットヘッダのptsからこのフィールドpresentation\_end\_time\_stampで示される有効期間終了時刻までである。フィールドnumber\_of\_Dispcmdsは、データ長が8ビットの0以上の整数で、ブロックGOBJCommandTable()に格納されている表示制
- 5 御命令の数を表す。フィールドnumber\_of\_PNG\_imagesは、データ長が8ビットの0以上の整数で、ブロックPNGImageRegion()に格納されているPNG画像の数を表す。フィールドnumber\_of\_sound\_dataが配置され、ブロックSoundDataRegion()に格納されている音声データの数がデータ長が8ビットの0以上の整数で示される。
- 10 ブロックGraphicsObjectHeader()内のブロックGlobalPaletteTable()は、このグラフィクスオブジェクト200で共通に使用されるパレットテーブルの情報が格納される。例えばこのブロックGlobalPaletteTable()に記述されるパレットテーブルの情報が、例えば共通パレットテーブル230の内容としてセットされる。フィールドstart\_addr
- 15 ess\_of\_PNG\_image(i)は、データ長が32ビットの0以上の整数で、i番目のPNG画像のデータPNG\_image(i)が開始される位置を、このGraphicsObject()の先頭からの相対バイト数で表現する。
- フィールドPNG\_file\_name(i)は、フィールドstart\_address\_of\_PNG\_image(i)から始まるPNGデータのファイル名を表す。ブロックPNG
- 20 ImageRegion()内のフィールドであるフィールドPNG\_image(i)の内容は、単体のPNGファイルと同一であり、ブロックPNGImageRegion()は、一つ以上のPNGファイルを連結して作られる。例えば、上述の第76図Aでは、PNGデータ203A、203B、・・・、203nが連結されてブロックPNGImageRegion()が作られる。その際にファ
- 25 イル名が失われないよう、フィールドPNG\_file\_name(i)にファイル名を保持しておくことができる。逆に、ブロックPNGImageRegion()を分

解して個別のPNGファイルを復元する際には、各フィールドPNG\_image(i)を、iが対応するフィールドPNG\_file\_name(i)で示されるファイル名の、独立したファイルとすればよい。

フィールドstart\_address\_of\_sound\_data(i)は、データ長が32ビットの0以上の整数で、i番目の音声データsound\_data(i)が開始される位置を、このブロックGraphicsObject()の先頭からの相対バイト数で表現する。

ブロックGOBJCommandTable()は、実行時刻が同じ表示制御命令を集めた命令グループDispCmds(i)から構成される。命令グループDispCmds(i)は、必ず実行時刻を指定する命令execution\_time(time)から開始して、表示制御命令が並べられる。換言すると、命令execution\_time(time)から次の命令execution\_time(time)の前までが一つの命令グループDispCmds(i)を構成している。

ブロックPNGImageRegion()は、上述したように、PNG方式で圧縮符号化された画像一枚分のデータであるフィールドPNG\_image(i)の並びで構成される。

ブロックSoundDataRegion()は、音声データsound\_data(i)として実際の音声データが格納される。

ブロックGraphicsObjectHeader()とブロックGOBJCommandTable()との間には、任意の数のpadding\_wordを挿入することができる。同様に、ブロックGOBJCommandTable()とブロックPNGImageRegion()の間には任意の数のpadding\_wordを挿入することができる。

第84図は、上述したブロックGlobalPaletteTable()の一例の構造を表すシンタクスを示す。フィールドnumber\_of\_palette\_entriesは、次に羅列されているパレットデータの数を表す。データ長が8ビットのインデックス番号で画像を表現する場合、このフィールドnumber

\_of\_palette\_entriesの値は、最大で256となり、256色を用いることができる。この256色のうち一部の色だけを使用する場合、必要なパレットデータだけが存在すればよい。このフィールドnumber\_of\_palette\_entriesにより、用いるインデックスの数が示される。

- 5     フィールドpalette\_index\_numberは、次のフィールドred\_value、フィールドgreen\_value、フィールドblue\_valueおよびフィールドalphaでそれぞれ定義される色や透明度に割り当てられたインデックス番号を表す。画像データからは、このインデックス番号によって色や透明度 $\alpha$ が参照される。
- 10    このブロックGlobalPaletteTable()においてfor文で示されるループの中では、同じ値を持つフィールドpalette\_index\_numberが二度以上現れてはならない。フィールドred\_value、フィールドgreen\_valueおよびフィールドblue\_valueは、それぞれデータ長が8ビットの0以上の整数であって、それぞれ赤、緑、青の色を指定する。8ビットの
- 15    データ長を有するフィールドalphaは、透明度 $\alpha$ を表し、値0が完全に透明、値255が完全に不透明な状態を表す。

- 各PNG画像は、パレット情報PLTEを格納するチャンクを持つことができるが、この実施の一形態では、そのパレット情報PLTEを使用せず、ブロックGlobalPaletteTable()で定義されたパレット情報
- 20    報を使うものとする。これは、同時に複数のPNG画像を表示する場合、各PNG画像が異なるパレットテーブルに基づく色を使っていると、それぞれ正しい色で表示することが困難になるためである。GraphicsObject()に属する、フィールドPNG\_image(i)でそれぞれ表される複数のPNG画像は、全て共通のブロックGlobalPaletteTable()を参
- 25    照し、ブロックGlobalPaletteTable()に示される共通のパレットテーブルを用いるものとする。

命令グループDispCmds(i)について説明する。命令グループDispCmds(i)は、グラフィクスオブジェクト200による表示を制御する表示制御命令が羅列される。命令グループDispCmds(i)において、命令execution\_time(start\_time)は、次の命令execution\_time(start\_time)の  
 5 前までの命令を、指定の時刻start\_timeに実行することを指示する。時刻start\_timeの原点は、当該グラフィクスオブジェクト200のptsとする。また時刻start\_timeの単位は、ptsと同じものが用いられる。

一つの命令グループDispCmds(i)は、命令execution\_time(start\_time)により時刻start\_timeから始まり、その他の複数の命令を置くことができる。命令グループDispCmds(i)に置かれたこれらの命令は、命令execution\_time(start\_time)で指定される時刻start\_timeに同時に、実行されるものとする。命令グループDispCmds(i)による命令の実行完了時刻前に、次の命令グループDispCmds(i+1)の命令execution\_time(start\_time)で示される時刻start\_timeになった場合は、命令  
 10 グループDispCmds(i)の実行は中断され、次の命令グループDispCmds(i+1)が実行される。

命令グループDispCmds(i)に置かれる、命令execution\_time(start\_time)以外の表示制御命令としては、第85図Aおよび第85図Bに一覧が例示されるように、以下のものが考えられる。各行頭の番号は、第85図の各番号にそれぞれ対応している。(すなわち、番号(1)を、命令execution\_time(start\_time)とする)

- (2) グラフィクスオブジェクトの表示を開始する命令
- (3) グラフィクスオブジェクトの表示を停止する命令
- 25 (4) 使用中のパレットテーブルの色や透明度を変更する命令
- (5) グラフィクスオブジェクトのプレーン上での表示位置やサイズ

を設定する命令

(6) グラフィクスオブジェクトの中の表示範囲を設定する命令

(7) 効果音の再生を行う命令

(8) 画像データ (PNGデータ) に効果音を割り当てる命令

- 5     これら `execution_time(start_time)` の後に置かれる 7 種類の命令は、一例であって、命令グループ `DispCmds(i)` に置かれる命令は、この 7 種類に限定されるものではない。表示制御命令をさらに定義し追加することは可能である。

- (2) および (3) の、グラフィクスオブジェクト 200 の表示の  
10    開始および終了命令は、所謂フェードイン／フェードアウトを行うための命令であって、それぞれ命令 `fade_in(fade_in_time)`、命令 `fade_out(fade_out_time)` の形で用いられる。

- フェードインは、命令 `fade_in(fade_in_time)` で指示され、グラフィクスオブジェクト 200 が非表示の状態から徐々に表示の状態に変化していく。アルファブレンディングの透明度  $\alpha$  の値を、時刻 `fade_in_time` に基づき徐々に増加させていくことで、フェードインが実現できる。命令 `execution_time(start_time)` の後に命令 `fade_in(fade_in_time)` が置かれていると、命令 `execution_time(start_time)` で指定される時刻 `start_time` の直後から当該グラフィクスオブジェクト 200  
15    が徐々に透明から不透明になっていき、引数の時刻 `fade_in_time` で指定された時間が経過すると、全てのパレットインデックスの透明度  $\alpha$  の値が共通パレットテーブルで指定した値に設定される。

- 命令 `fade_in(fade_in_time)` において、時刻 `fade_in_time` が 0 に設定されていると、当該グラフィクスオブジェクト 200 がパレット  
20    テーブルに指定される色および透明度  $\alpha$  で即座に表示される。

フェードアウトは、フェードインの逆の処理であって、命令 `fade_o`



ut(fade\_out\_time)で指示され、グラフィクスオブジェクト200が表示の状態から徐々に非表示の状態に変化していく。アルファブレンディングの透明度 $\alpha$ の値を、時刻fade\_out\_timeに基づき徐々に減少させていくことで、フェードアウトが実現できる。命令execution\_time(5 start\_time)の後に命令fade\_out(fade\_out\_time)が置かれていると、命令execution\_time(start\_time)で指定される時刻start\_timeの直後から当該グラフィクスオブジェクト200が徐々に不透明から透明になっていき、引数の時刻fade\_out\_timeで指定された時間が経過すると、全てのパレットインデックスの透明度 $\alpha$ の値が0とされ、当該グラフィクスオブジェクト200が完全に透明になり、見えなくなる。10

命令fade\_out(fade\_out\_time)において、時刻fade\_out\_timeが0に設定されていると、当該グラフィクスオブジェクト200の表示が即座に停止される。

15 フェードインおよびフェードアウトにおいて、透明度 $\alpha$ の値は、時間の経過と共にできるだけ滑らかに変化させると、より自然なフェードインおよびフェードアウトの効果が得られ、好ましい。これは、この例に限られず、例えば、フェードインにおいて、透明度 $\alpha$ の値は、時刻fade\_in\_timeで示される時間が経過した後に、パレットテーブル20で指定した値になっていればよく、どの程度の分解能や階調で透明度 $\alpha$ を変化させるかについては、命令によって指定していない。実際には、透明度 $\alpha$ の変化の分解能や階調は、実装に依存する。

ここでは可読性を上げるために、命令を「fade\_in()」、「fade\_out()」のようにテキストで表記して説明したが、実際には、これらの25 命令fade\_in()、fade\_out()は、その引数と共に所定のバイナリ値に変換されてDispCmds(i)に記述される。これは、以降に説明される他

の命令についても同様である。

(4) の、使用中のパレットテーブルの色や透明度  $\alpha$  を変更する命令は、パレット情報を変更する。この命令は、命令 `change_palette(index, newR, newG, newB, newAlpha)` の形で用いられる。字幕プレーン 1 1 やグラフィクスプレーン 1 2 に同時に表示される PNG 画像は、第 8 3 図に示したシンタクスで定義される、第 2 4 図のような共通のパレットテーブルを参照している。通常は、`GlobalPaletteTable()` で定義されたパレット情報がそのまま共通パレットテーブルとして使用される。この命令 `change_palette(index, newR, newG, newB, newAlpha)` を用いることで、この共通のパレット情報を変更することができる。

命令 `change_palette(index, newR, newG, newB, newAlpha)` に引数として記述される値 `index`、`newR`、`newG`、`newB` および `newAlpha` により、パレット 2 2 において、パレット番号 `index` で示されるカラーインデックス値の三原色の値 `R`、`G` および `B` が値 `newR`、`newG` および `newB` にそれぞれ変更され、透明度  $\alpha$  の値が値 `newAlpha` に変更される。

(5) の、グラフィクスオブジェクトのプレーン上での表示位置やサイズを設定する命令は、命令 `set_display_box(x1, y1, x2, y2)` の形で用いられ、プレーン上の座標 `(x1, y1)` および `(x2, y2)` で定められる矩形領域 `(x1, y1) (x2, y2)` に、当該グラフィクスオブジェクト 2 0 0 を配置する。また、(6) の、グラフィクスオブジェクトの中の表示範囲を設定する命令は、命令 `set_clipping_box(a1, b1, a2, b2)` の形で用いられ、グラフィクスオブジェクト 2 0 0 による PNG 画像上の座標 `(a1, b1)` および `(a2, b2)` で定められる矩形領域 `(a1, b1) (a2, b2)` を、実際にプレーン上に表示する。

第 8 6 図 A および第 8 6 図 B を用いて、命令 `set_display_box(x1, y`

1, x2, y2) および命令 `set_clipping_box(a1, b1, a2, b2)` について、より具体的に説明する。第 8 6 図 A および第 8 6 図 B における座標は、第 8 7 図に一例が示されるように、表示画面の左上隅を原点とし、水平右方向を x、垂直下方向を y として、座標 (x, y) と表す。

5 命令 `set_clipping_box(a1, b1, a2, b2)` により、第 8 6 図 A に示されるように、グラフィクスオブジェクト 2 0 0 による PNG 画像 2 5 0 内の、実際に表示される矩形領域 (a1, b1) (a2, b2) が設定される。この第 8 6 図 A の例では、設定される矩形領域 (a1, b1) (a2, b2) が PNG 画像 2 5 0 に対して小さいものとされている。命令 `set_display_box(x1, y1, x2, y2)` により、この矩形領域 (a1, b1) (a2, b2) のプレーン上における実際の表示位置が矩形領域 (x1, y1) (x2, y2) に設定される (第 8 6 図 B 参照)。画面上の矩形領域 (x1, y1) (x2, y2) に対して、PNG 画像 2 5 0 のうち矩形領域 (a1, b1) (a2, b2) の部分が表示される。

矩形領域 (a1, b1) (a2, b2) が実際に表示される矩形領域 (x1, y1) (x2, y2) よりも大きい場合は、矩形領域 (a1, b1) (a2, b2) のうち矩形領域 (x1, y1) (x2, y2) の範囲の PNG 画像だけが表示される。一方、矩形領域 (a1, b1) (a2, b2) の大きさが実際に表示される矩形領域 (x1, y1) (x2, y2) よりも小さい場合は、矩形領域 (x1, y1) (x2, y2) 内の矩形領域 (a1, b1) (a2, b2) の外側は、透明色として扱うとよい。

20 上述した表示制御命令を、命令 `execution_time(start_time)` を複数用いて並べることにより、時間経過に従って変化する字幕や同期型グラフィクスを表示することができる。例えば、第 8 3 図を用いて説明したグラフィクスオブジェクト 2 0 0 において、ブロック `G0BJCommandTable()` 内に、複数の命令グループ `DispCmds(i)` を記述する。そして、命令グループ `DispCmds(i)` のそれぞれは、命令 `execution_time(start_time)` における時刻 `start_time` を所望に異ならせて記述し、時刻 `sta`

rt\_timeで示される時刻に開始する表示制御命令を記述する。

第88図は、命令グループDispCmds(i)の記述とグラフィクスオブジェクト200の表示変化の一例を示す。第88図において、横軸は時間の経過を表し、縦軸はグラフィクスオブジェクト200の透明度を表す。透明度は、第88図で上に向かうほど大きくなる。ptsで示される時刻が原点とされる。

最初の命令グループDispCmds(0)により、プレーン上の表示領域が命令set\_display\_box(800,800,1300,900)として、グラフィクスオブジェクト200のPNG画像における表示領域が命令set\_clipping\_box(0,0,500,100)として設定し、命令fade\_in(2 sec)により、2秒かけてフェードインする処理を、時刻〔0〕から開始することが指示される。次の命令グループDispCmds(1)は、命令change\_palette(index, newR, newG, newB, Alpha)がカラーインデックス値〔1〕、〔2〕〔3〕および〔4〕についてそれぞれ記述され、時刻〔800〕から、パレットテーブル22においてカラーインデックス値〔1〕、〔2〕〔3〕および〔4〕で参照される色および透明度 $\alpha$ を変更する旨が指示される。そして、次の命令グループDispCmds(2)により、表示されているグラフィクスオブジェクト200を、時刻〔2000〕から2秒かけてフェードアウトすることが指示される。

この第88図に示されるように、命令グループDispCmds(0)、DispCmds(1)およびDispCmds(2)を並べることで、時間経過に従って表示が変化する例えば字幕を実現することができる。命令グループDispCmds(0)、DispCmds(1)およびDispCmds(2)を適当に用いることで、字幕やボタン画像のアニメーション表示が可能とされる。

第89図A、第89図B、第89図Cおよび第89図Dは、字幕表示が徐々に現れるフェードインの例を示す。第89図A～第89図D

と、徐々に字幕表示が現れるように表示制御される。このような単純なフェードインは、上述した第 88 図の命令グループ DispCmds (0) と同様の命令で実現することができる。

第 90 図 A および第 90 図 B は、字幕の PNG 画像 260 がプレーン内を移動する例を示す。これは、命令 set\_display\_box(x1,y1,x2,y2) を複数、用いることで実現できる。例えば、最初の命令グループ DispCmds (0) において、命令 execution\_time(start\_time) により開始時刻が設定され、命令 set\_clipping\_box(a1,b1,a2,b2) により第 90 図 A に一例が示されるように PNG 画像 260 の表示領域を設定すると共に、命令 set\_display\_box(x1,y1,x2,y2) により、PNG 画像 260 のプレーン内での初期の表示領域を設定する。

次の命令グループ DispCmds (1) において、命令 execution\_time(start\_time) により命令グループ DispCmds (0) から所定時間経過した時刻が開始時刻として指定されると共に、命令 set\_display\_box(x1',y1',x2',y2') により、プレーン内の移動先の表示領域が設定される。同様に、次の命令グループ DispCmds (2) において、命令 execution\_time(start\_time) により命令グループ DispCmds (1) から所定時間経過した時刻が開始時刻として指定されると共に、命令 set\_display\_box(x1'',y1'',x2'',y2'') により、プレーン内の移動先の表示領域が設定される。

20    このようにすることで、第 90 図 B に一例が示されるように、字幕の PNG 画像 260 がプレーン上の矩形領域 (x1,y1) (x2,y2)、矩形領域 (x1',y1') (x2',y2')、矩形領域 (x1'',y1'') (x2'',y2'') と移動することができる。

第 91 図 A および第 91 図 B は、字幕の PNG 画像 261 内の表示領域 262 が移動し、字幕表示をスクロールさせる例を示す。これは、命令 set\_clipping\_box(a1,b1,a2,b2) を複数、用いることで実現で

きる。例えば、最初の命令グループDispCmds(0)において、命令execution\_time(start\_time)により開始時刻が設定され、命令set\_clipping\_box(a1,b1,a2,b2)により、第91図Aに一例が示されるように、PNG画像260において初期に表示される矩形領域262を設定すると共に、命令set\_display\_box(x1,y1,x2,y2)により、PNG画像260がプレーン内で表示される矩形領域を設定する。

次の命令グループDispCmds(1)において、命令execution\_time(start\_time)により命令グループDispCmds(0)から所定時間経過した時刻が開始時刻として指定されると共に、命令set\_clipping\_box(a1',b1',a2',b2')により、PNG画像260内の移動先の表示領域が設定される。同様に、次の命令グループDispCmds(2)において、命令execution\_time(start\_time)により命令グループDispCmds(1)から所定時間経過した時刻が開始時刻として指定されると共に、命令set\_clipping\_box(a1'',b1'',a2'',b2'')により、PNG画像260内の移動先の矩形領域が設定される。

このようにすることで、第91図Bに一例が示されるように、プレーン上の矩形領域(x1,y1)(x2,y2)内を、字幕のPNG画像261内の一部の矩形領域が矩形領域(a1,b1)(a2,b2)、矩形領域(a1',b1')(a2',b2')、矩形領域(a1'',b1'')(a2'',b2'')と移動するようにされ、字幕表示のスクロールを実現できる。

第92図Aおよび第92図Bは、PNG画像265の一部を表示する枠を設定し、PNG画像265上でこの枠を移動させつつ、プレーン上での表示位置も移動させる例を示す。これは、命令set\_display\_box(x1,y1,x2,y2)と、命令set\_clipping\_box(a1,b1,a2,b2)とを複数、同時に用いることで実現できる。例えば、最初の命令グループDispCmds(0)において、命令execution\_time(start\_time)により開始時刻

が設定され、命令set\_display\_box(x1,y1,x2,y2)と命令set\_clipping\_box(a1,b1,a2,b2)とにより、枠266Aが設定される（第92図A参照）。

例えば、命令set\_display\_box(x1,y1,x2,y2)によりプレーン上で表示される矩形領域(x1,y1)(x2,y2)が設定されると共に、命令set\_clipping\_box(a1,b1,a2,b2)によりPNG画像265において表示される矩形領域(a1,b1)(a2,b2)が設定される。これら矩形領域(x1,y1)(x2,y2)および矩形領域(a1,b1)(a2,b2)により、枠266Aが設定される。

そして、次の命令グループDispCmds(1)において、命令execution\_time(start\_time)により命令グループDispCmds(0)から所定時間経過した時刻が開始時刻として指定され、命令set\_display\_box(x1',y1',x2',y2')によるプレーン内の矩形領域(x1',y1')(x2',y2')と、命令set\_clipping\_box(a1',b1',a2',b2')によるPNG画像265上の矩形領域(a1',b1')(a2',b2')により、枠266Aに対する移動先の枠266Bが設定される。同様に、次の命令グループDispCmds(2)において、命令execution\_time(start\_time)により命令グループDispCmds(1)から所定時間経過した時刻が開始時刻として指定され、命令set\_display\_box(x1'',y1'',x2'',y2'')によるプレーン内の矩形領域(x1'',y1'')(x2'',y2'')と、命令set\_clipping\_box(a1'',b1'',a2'',b2'')によるPNG画像265上の矩形領域(a1'',b1'')(a2'',b2'')により、枠266Bに対する移動先の枠265Bが設定される。

こうすることで、第92図Bに一例が示されるように、字幕のPNG画像265内の一部の矩形領域が移動しつつ、当該矩形領域がプレーン内を領域265A、265B、265Cと移動するような表示が可能となる。

このように、この発明の実施の一形態では、グラフィクスオブジェ

クト 200 の表示制御を、命令 `execution_time(start_time)` により各表示制御命令をグループ化した命令グループ `DispCmds(i)` を用いて行っているため、字幕プレーン 11 およびグラフィクスプレーン 12 における多彩な表示を、容易に実現することができる。

5

### 2-13. 効果音について

この発明の実施の一形態では、さらに、グラフィクスオブジェクト 200 の表示制御に同期して音声を出力することができる。上述した、命令グループ `DispCmds(i)` に置かれる、命令 `execution_time(start`  
10 `_time)` 以外の表示制御命令 (2) ~ (8) のうち、(7) の効果音の再生を行う命令と、(8) の画像データに効果音を割り当てる命令により、音声出力が定義される。また、音声データには、それぞれ識別子 `sound_id` が付与され、互いが識別される。

(7) の、効果音の再生を行う命令は、命令 `play_sound(sound_id)`  
15 の形で用いられ、識別子 `sound_id` で指定された音声データの再生を指示する。この命令を命令グループ `DispCmds(i)` に含めることにより、命令 `execution_time(start_time)` で指定された時刻 `start_time` に、識別子 `sound_id` で指定された音声データが再生される。

一例として、命令 `play_sound(sound_id)` を命令 `fade_in(fade_in_time)`  
20 `me)` および命令 `fade_out(fade_in_time)` と同時に使用することで、字幕の表示および／または消去に連動して音声データを再生し、効果音を発生させることが可能となる。第 88 図は、命令 `play_sound(sound_id)` を使用した例である。第 88 図の例では、最初の命令グループ `DispCmds(0)` において、開始時刻 [0] で 2 秒かけて表示がフェードイ  
25 `ン` すると共に、命令 `play_sound(1)` により識別子 `sound_id` が [1] で特定される音声データが再生される。次いで、命令グループ `DispCmds`



(1)において、時刻〔800〕で表示色が変更されると共に、命令play\_sound(2)により識別子sound\_idが〔2〕で特定される音声データが再生される。そして、命令グループDispCmds(2)において、時刻〔2000〕から1秒かけて表示がフェードアウトすると共に、命令play\_sound(1)により識別子sound\_idが〔1〕で特定される音声データが再生される。

命令play\_sound(sound\_id)は、必須の命令ではない。

(8)の、PNGデータに効果音を割り当てる命令は、命令set\_sound(PNG\_image\_id, sound\_id)の形で用いられ、識別子PNG\_image\_idで指定されるPNGデータに対して、識別子sound\_idで指定される音声データを割り当てる。この命令set\_sound(PNG\_image\_id, sound\_id)により、識別子PNG\_image\_idで指定されるPNGデータが表示される際に、識別子sound\_idで指定される音声データが再生される。PNGデータの識別子PNG\_image\_idは、例えば、上述した第83図における、ブロックPNGImageRegion()中のPNG\_image(i)に対して指定されるループカウンタiと同じ値を用いることができる。

この命令set\_sound(PNG\_image\_id, sound\_id)は、主に、選択状態および実行状態のボタンのPNGデータに用いることが想定されている。こうすることで、ボタンが通常状態から選択状態へ遷移する際や、選択状態から実行状態に遷移する際に、それぞれの状態を表すPNGデータに割り当てられた音声データを効果音として発生させることが可能となる。この例に限らず、ボタン以外のPNGデータに対してこの命令set\_sound(PNG\_image\_id, sound\_id)を用いてもよい。

第93図は、ボタン画像に音声データを割り当てたグラフィクスオブジェクト200の一例のデータ構造を示す。PNGデータ領域203に、通常状態、選択状態および実行状態それぞれにおけるボタンの

PNGデータ203A、203Bおよび203Cが格納される。この第93図の例では、表示制御命令ではPNGデータに対する座標指定と、音声データの割り当てのみを行い、PNGデータの表示開始時刻や、表示開始時のボタンの初期状態は、外部のスクリプトプログラムにより制御するようにしている。そのため、表示制御命令は、便宜上、実行時刻〔0〕の命令として記述する。

第93図に例示されるグラフィクスオブジェクト200では、表示制御命令テーブル202において、命令execution\_time(0)に基づき時刻〔0〕で実行される、命令グループDispCmds(0)だけが存在している。識別子PNG\_image\_idは、〔0〕から始まるので、識別子PNG\_image\_idが〔0〕で通常状態時のPNGデータ203Aを、識別子PNG\_image\_idが〔1〕で選択状態時のPNGデータ203Bを、識別子PNG\_image\_idが〔2〕で実行状態時のPNGデータ203Cをそれぞれ表している。

15 命令set\_sound(1,10)に基づき、識別子PNG\_image\_idが〔1〕である選択状態のボタンを表示するPNGデータ203Bが表示されたときには、識別子sound\_idが〔10〕で指定される音声データが効果音として再生される。同様に、命令set\_sound(2,11)に基づき、識別子PNG\_image\_idが〔2〕である実行状態のボタンを表示するPNGデータ203Cが表示されたときには、識別子sound\_idが〔11〕で指定される音声データが効果音として再生されることになる。

第73図A、第73図Bおよび第73図Cでは省略されているが、プレーヤは、予め1または複数種類の音声データを、プレーヤ内部のメモリなどに保持することができる。例えば、プレーヤは、プレーヤ内部に設けられた不揮発性メモリなどに、出荷時に予め所定の音声データを記憶しておくようにできる。

これに限らず、グラフィックオブジェクト 200 や動画データによるコンテンツが記録されたディスクに、効果音として用いるための音声データを記録しておき、当該ディスクの再生時にこの音声データを読み込んでおくようにしてもよい。ディスクへの音声データの格納方法としては、音声データが格納されたファイルを別に用意し、ディスク再生開始時にこのファイルを先に読み込んでおき、プレーヤ内のメモリに置いておく方法がある。

グラフィクスオブジェクト 200 と同様に音声データを格納した PES パケットを作り、それを TS パケットに分割してクリップ AV ストリームに多重化する方法も考えられる。

上述の第 76 図 A、第 76 図 B および第 76 図 C のグラフィクスオブジェクト 200 中のグラフィクスオブジェクトヘッダ 201 内（図示は省略する）、或いは、上述したように PNG 画像データ領域 203 の直後に音声データ領域 204 を設け、グラフィクスオブジェクト 200 に含まれる PNG 画像に対応する音声データを格納する方法も考えられる。

何れの方法であっても、音声データを予めディスクからプレーヤのメモリに読み込んで蓄積しておくことができるので、PNG 画像で作られたボタンが選択状態や実行状態になると同時に効果音を発生させることが可能となる。音声データには、それぞれユニークな識別子 sound\_id が割り当てられており、音声データを一意に特定することができるようにされている。

この、音声データのディスクへの格納方法について、より詳細に説明する。グラフィクスオブジェクト 200 に音声データを格納する方法について、第 76 図 A、第 76 図 B および第 76 図 C を用いて説明する。第 76 図 A および第 76 図 C は、音声データをグラフィクスオ

プロジェクト 200 に付加してクリップ AV ストリームに多重化する例が示されている。

第 76 図 A は、グラフィクスオブジェクト 200 中の PNG データ領域 203 の後ろに音声データ領域 204 を設けた例である。音声データ領域 204 には、複数の音声データ 204 A、204 B、・・・、204 n を格納することができる。これら音声データ 204 A、204 B、・・・、204 n を、同一のグラフィクスオブジェクト 200 に格納される PNG データ 203 A、203 B、・・・、203 n とそれぞれ関連性のあるものとする、PNG 画像および音声データを対応付けての扱いが容易になる。

音声データ 204 A、204 B、・・・、204 n は、例えば A I F F (Audio Interchange File Format) 形式や W A V E ファイルといった、圧縮符号化されていないデータでもよいし、M P 3 (Moving Pictures Experts Group 1 Audio Layer 3) や A A C (Advanced Audio Coding)、A T R A C (Adaptive Transform Acoustic Coding) といった、圧縮符号化されたデータでもよい。圧縮符号化された音声データを格納する場合には、プレーヤ側が音声データの圧縮符号化方式に対応したオーディオデコーダを持っている必要がある。

第 76 図 C は、ボタンを構成するグラフィクスオブジェクト 200 に対して、ボタンのそれぞれの状態に対応した音声データを格納した例である。この例では、例えばボタンの選択状態時に再生する音声データ 204 A-2 と、実行状態時に再生する音声データ 204 B-2 とが、音声データ領域 203 に格納される。一方、PNG データ領域 203 には、通常状態、選択状態および実行状態それぞれのボタン表示に用いる PNG データ 203 A-2、203 B-2 および 203 C-2 が格納される。

この場合、例えば、ボタン画像が格納されるPNGデータ領域203に続く音声データ領域204には、ボタンが選択状態時に遷移したときに再生する音声データ204A-2と、実行状態時に遷移したときに再生する音声データ204B-2とが格納されるように、予め決めておく。そして、ボタン画像のPNGデータが表示されると同時に、当該ボタンの状態に対応した音声データが再生されるようにする。このプレーヤで再生される効果音は、主にボタンクリック音として使われることを想定しているため、このようなルールでも、十分に、主要な目的を達成することができる。

10 音声データをクリップAVストリームに多重化しない方法について説明する。例えば、第94図に一例が示されるように、ディレクトリBD AVの下に音声データを格納するディレクトリSOUNDを設ける。このディレクトリSOUNDには、音声データであるPCM波形データを格納する、例えばA I F F形式の音声データファイル「soundl.aiff」を置く。ディレクトリSOUNDに格納された音声データ

15 ファイルは、例えば当該ディスクのプレーヤへの最初のローディング時にまとめて読み込まれ、プレーヤの内部メモリに格納される。

各音声データには、音声データそれぞれを識別可能な識別子sound\_idが予め割り当てられ、プログラムやスクリプトなどから識別子sound\_idにより、所望の音声データがメモリから呼び出されるモデルが構成される。

20

この場合、例えば、第95図に一例が示されるように、グラフィクスオブジェクト200に対してサウンドid領域205を設け、このサウンドid領域205に対してサウンドidデータ205A、205Bを格納する。この第95図の例では、PNGデータ領域203にボタンの通常状態時、選択状態時および実行状態時にそれぞれ対応し

25

- たPNGデータ203A、203Bおよび203Cが格納され、サウンドidデータ205Aおよび205Bは、PNGデータ203Bおよび203Cにそれぞれ対応する識別子sound\_idである。PNGデータ203Bの表示の際に、プレーヤ内部のメモリに予め格納された、
- 5 サウンドidデータ205Aが示す識別子sound\_idが対応する音声データを再生するように指示される。

例えば、第93図を用いて既に説明したように、表示制御命令テーブル202の記述によって、PNGデータと音声データとを関連付けるようにしてもよい。

- 10 この第95図に一例が示される構成のデータは、ディスクから読み出すだけでなく、インターネットなどのネットワークから取得し、入力端101に入力するようにもできる。

- 識別子sound\_idにより音声データを指定するため、第85図を用いて既に説明した表示制御コマンドなどにより、グラフィクス表示とは
- 15 関係の無い任意の時刻に、音声データによる効果音などを発生させることができる。

- この方法では、識別子sound\_idで音声データをメモリから呼び出すため、効果音の種類は、第1に、識別子sound\_idの数で制限される。第2に、使える効果音の種類が、上述したように、プレーヤの内部メモリ
- 20 の容量により制限される。

- 第96図を用いて、より具体的に説明する。ディスク400のローディング時に、プレーヤによりディスクに対してイニシャルアクセスがなされ、ディレクトリBD A Vの下に配置されるディレクトリS O U N Dから、音声データがまとめて読み込まれる。読み込まれた音声
- 25 データ（PCMデータ）は、プレーヤの内部メモリ410に格納される。このとき、音声データのそれぞれに対して固有の識別子sound\_id

が割り当てられる。ディスク 400 に記録されている音声データに対して予め識別子 sound\_id を付加しておいてもよい。

この例では、16 個の音声データがディスク 400 から読み込まれ、識別子 sound\_id=1~16 がそれぞれの音声データに割り当てられる。

- 5 音声データそれぞれのデータサイズが取得される。この第 96 図の例では、識別子 sound\_id=1~16 の音声データがそれぞれ d1 バイト、d2 バイト、・・・、d16 バイトのデータサイズを有しているものとする。

- 例えば、ボタン 420A、420B および 420C が表示されるメニュー画面 420 における、ボタン 420C に対する操作に基づき、ボタン 420C に対して指定された識別子 sound\_id により対応する音声データがメモリ 410 から読み出される。この第 96 図の例では、ボタン 420C の実行状態に対して識別子 sound\_id=1 で示される音声データが指定されている。メモリ 410 から読み出されたこの音声データは、所定に処理されてバッファ 450B に一旦溜め込まれる。そして、オーディオミキサ 231 に供給され、例えば本編の動画像データに付随する音声データと混合されて、音声として出力される。
- 10
- 15

- バッファ 450A には、例えば本編の動画像データに付随する音声データが一旦、溜め込まれる。例えば、バッファ 450A および 450B の間で、溜め込まれた音声データを読み出すタイミングを調整することで、バッファ 450B に溜め込まれたボタン 420C の操作に対応した効果音がバッファ 450A に溜め込まれた音声データの適当なタイミングで出力される。この例では、識別子 sound\_id=0 で、音声データの再生無しが指定される。
- 20

- 25 このようなモデルにおいて、ディスク 400 から読み込み可能な音声データの合計容量がメモリ 410 の容量に基づき制限される。それ

ぞれの音声データの容量がバッファ 450 B の容量に基づき制限される。メモリ 410 の容量を容量  $M$  (バイト)、バッファ 450 B の容量を容量  $D_{max}$  (バイト) とした場合、次の 2 つの条件を満たす必要がある。

5 (1) メモリ 410 に格納された音声データのそれぞれの容量  $d_i$  がバッファ 450 B の容量  $D_{max}$  より小さい。

(2) メモリ 410 に格納された音声データの総容量 ( $d_1 + d_2 + \dots + d_n$ ) がメモリ 410 の容量  $M$  より小さい。

換言すれば、この (1) および (2) の条件をプレーヤ側とディスク制作側とに対してルール化し、規格として定めておくことで、効果音などの音声データの再生に関して再生互換性を確保することができる。

音声データをクリップ AV ストリームに多重化しない場合に、音声データをディスクのプレーヤへの最初のローディング時にまとめて読み込むように説明したが、これはこの例に限定されない。一例として、音声データを複数回に分けて読み出すようにできる。例えば、シナリオの進行に応じてある区切となる箇所、次の区切りまでに用いるだけの音声データをまとめて読み出し、メモリに記憶させる。このとき、それまでメモリに記憶されていた音声データを消去する。こうすることで、1 つのシナリオの再生において、メモリの容量以上の音声データを扱うことが可能とされる。

音声データは、ディスクの所定領域にまとめて記録しておいてもよいし、ディスクの複数領域に分けて記録しておいてもよい。ディスクの複数領域に分けて記録する場合には、例えば、シナリオの進行に応じて、シナリオの区切りに対応したディスク上の位置に、次の区切りまでに用いられる音声データをまとめて記録することが考えられる。



ネットワークで結ばれたサーバーから音声データをダウンロードすることも、ファイルが置かれている場所を例えばURL (Uniform Resource Locator) で指定するなどすれば、ディスクからの読み出しと同様の仕組みで実現可能である。

- 5 第76図Aおよび第76図Bを用いて説明した、音声データをクリップAVストリームに多重化する方法では、音声データの種類の制約が無く、1画像毎に異なる音声データを指定できる利点がある。音声データは、必要に応じてクリップAVストリームにより供給されるため、その都度異なる音声データを用いることができる。音声データを
- 10 クリップAVストリームに多重化する方法では、音声データが画像データと同時にクリップAVストリームから読み込まれるため、読み込みモデルが簡単になる、音声データのファイル数の制限やファイルサイズの制限は、ディスク容量以外の制約を受けない、などの利点がある。
- 15 音声データをクリップAVストリームに多重化する方法では、異なるグラフィクスオブジェクトで同一の音声データを扱うような場合、それぞれのグラフィックオブジェクトが同一の音声データをそれぞれ持たなければいけないので、データが冗長になってしまう。グラフィクスオブジェクトから音声データを抽出する必要があるため、クリッ
- 20 プAVストリームのデマルチプレクス処理の後に、音声データを分離する処理が必要となる。

第73図A、第73図Bおよび第73図Cを用いてグラフィクスオブジェクト200に連動可能な音声データ処理を説明する。

- 上述の、クリップAVストリームに多重化されない音声データは、
- 25 例えば入力チャンネル(1)のデータとして入力端101に入力され、スイッチ回路102および103を介してコンテンツバッファ10

5 に供給される。一方、音声データが格納されたグラフィクスオブジェクト200が多重化されたクリップAVストリームは、入力端202に入力される。そして、PIDフィルタ110でグラフィクスオブジェクト200が振り分けられバッファTBn111Aに一旦溜め込まれ、スイッチ回路103を介してコンテンツバッファ105に供給される。

音声データの格納されていないグラフィクスオブジェクト200がクリップAVストリームに多重化されて入力端202から入力される。PIDフィルタ110でグラフィクスオブジェクト200を構成する10 トランスポートパケットが振り分けられ、バッファTBn111BまたはバッファTBn111Cに一旦溜め込まれる。バッファTBn111Bに溜め込まれたトランスポートパケットは、バッファBn112Bに供給され、PIDヘッダに基づきグラフィクスオブジェクト200がまとめられる。グラフィクスオブジェクト200は、スイッ15 チ回路107を介してグラフィクスデコーダA116に供給される。バッファTBn111Cに溜め込まれたトランスポートパケットも、同様にしてバッファBn112Cを用いてグラフィクスオブジェクト200がまとめられ、スイッチ回路108を介してグラフィクスデコーダB117に供給される。

20      グラフィクスデコーダA116およびB117では、供給されたトランスポートパケットのヘッダ情報を除去すると共に、当該トランスポートパケットに格納された画像データをデコードして字幕またはグラフィクスを表示するための画像データ、例えばビットマップデータとする。

25      コンテンツバッファ105に供給された、音声データが格納された  
グラフィクスオブジェクト200に格納される画像データは、コンテ

ンツバッファ 105 から取り出され、スイッチ回路 107 および 108 を介してグラフィクスデコーダ A 116 および B 117 にもそれぞれ供給される。

この第 7 3 図 A、第 7 3 図 B および第 7 3 図 C の例では、グラフィクスデコーダ A 116 は、字幕プレーンに展開する画像データをデコードし、グラフィクスデコーダ B 117 は、グラフィクスプレーンに展開する画像データをデコードする。グラフィクスデコーダ A 116 および B 117 は、これに限らず、さらに他のデータ形式の画像データに  
5 対応するようにもできる。

グラフィクスデコーダ A 116 の出力は、スイッチ回路 130 の入力端 130 B およびスイッチ回路 131 の入力端 131 C に供給され、スイッチ回路 130 および 131 を介してそれぞれ字幕プレーン 132 およびグラフィクスプレーン 133 に供給される。

マルチメディアエンジン 106 は、サウンドプレーヤ 106 D を有する。バッファ 109 は、サウンドバッファ 109 E を有する。サウンドプレーヤ 106 D は、サウンドバッファ 109 E を用いて、コンテンツツバッファ 105 から読み出した音声データを復号化して、例えば  
15 ニア PCM オーディオデータにして出力する。サウンドプレーヤ 106 D から出力された音声データは、プレゼンテーションプロセッサ 157 に供給され、オーディオデコーダ 118 から出力された音声データと混合されて、出力端 158 に導出される。

例えばボタン画像をクリックした際に発生されるクリック音など、効果音となる音声データは、サウンドプレーヤ 106 D によって再生  
25 される。音声データは、サウンドバッファ 109 E に蓄積され、サウンドプレーヤ 106 D により再生される。

マルチメディアエンジン 106 では、例えばシナリオ記述言語として HTML と E C M A スクリプトとを組み合わせ用いている場合コードバッファ 104 に蓄えられた E C M A スクリプトを読み出して解析し、コードバッファ 104 からの他の E C M A スクリプトや HTML 文書の読み出し、コンテンツバッファ 105 からの画像データや音声データの読み出しなどを行う。コンテンツバッファ 105 に格納された音声データは、コンテンツバッファ 105 に格納された他のデータと同様、コンテンツバッファ 105 に保持しておくことができる。

マルチメディアエンジン 106 により、ユーザからの、リモートコントロールコマンドやポインティングデバイスなどによる入力を受け取られ、その入力に基づく処理が行われる。マルチメディアエンジン 106 では、このユーザ入力に基づく処理結果や各スクリプトに応じてコントロール信号を生成する。このコントロール信号は、グラフィクスデコーダ A 116 および B 117、オーディオデコーダ 118、M P E G ビデオデコーダ 120 およびシステムデコーダ 121 にも供給される。

グラフィクスレンダラ 106 C で処理された画像データは、スイッチ回路 130 および 131 をそれぞれ介して字幕プレーン 132 およびグラフィクスプレーン 133 に供給される。字幕プレーン 132 およびグラフィクスプレーン 133 は、例えばフレームメモリからなり、第 20 図を用いて説明した字幕プレーン 11 およびグラフィクスプレーン 12 にそれぞれ対応する。

グラフィクスレンダラ 106 C から字幕プレーン 132 およびグラフィクスプレーン 133 に供給される画像データは、例えばランレングス圧縮形式、P N G 形式または J P E G 形式の画像データをグラフィクスレンダラ 106 C でデコードした後のビットマップデータであ

るものとする。

マルチメディアエンジン 106 は、プレゼンテーションプロセッサ 155 に対して、字幕プレーン 132、グラフィクスプレーン 133 および動画像プレーン 134 を切り換える制御信号を供給し、プレゼンテーションプロセッサ 141 に対して、オーディオストリーム出力を制御するような制御信号を供給する。

字幕プレーン 132 上の画像データは、第 22 図におけるパレット 22 に対応するパレット 150 に供給され、256 色からなるパレットに対してインデックスによる参照がなされ、RGB データが出力されると共に、透明度データ  $\alpha 1$  が出力される。RGB データは、第 22 図における RGB/YCbCr 変換回路 29 に対応する RGB/YCbCr 変換回路 151 に供給され、カラーシステムが RGB (4:4:4) から YCbCr (4:4:4) に変換される。RGB/YCbCr 変換回路 151 から出力された YCbCr データは、プレゼンテーションプロセッサ 155 に供給される。

グラフィクスプレーン 133 上の画像データは、第 22 図におけるパレット 26 に対応するパレット 152 に供給され、256 色からなるパレットに対してインデックスによる参照がなされ、RGB データが出力されると共に、透明度データ  $\alpha 2$  が出力される。RGB データは、第 22 図における RGB/YCbCr 変換回路 27 に対応する RGB/YCbCr 変換回路 153 に供給され、カラーシステムが RGB (4:4:4) から YCbCr (4:4:4) に変換される。RGB/YCbCr 変換回路 153 から出力された YCbCr データは、プレゼンテーションプロセッサ 155 に供給される。

プレゼンテーションプロセッサ 155 には、動画像プレーン 134 上の動画像データがアップ/ダウンコンバータ 154 を介して供給さ

れる。

プレゼンテーションプロセッサ 155 は、字幕プレーン 11 (字幕  
プレーン 132) による透明度  $\alpha 1$  と、グラフィクスプレーン 12 (  
グラフィクスプレーン 133) による透明度  $\alpha 2$  とを用いたアルファ  
5 ブレンディング処理を行う。この処理により、動画プレーン 10、字  
幕プレーン 11 およびグラフィクスプレーン 12 の画像データが合成  
される。プレゼンテーションプロセッサ 155 は、画像データに対し  
てリアルタイムでエフェクト処理を行うこともできる。このように、  
プレーン間の合成処理やエフェクト処理がなされた画像データが出力  
10 端 156 に導出される。

#### 2-14. その他

以上説明したように、この発明によれば、記録済み大容量ディスク  
において、動画を表示する動画プレーンと、字幕を表示する字幕プレ  
15 ーンと、メニュー画面などインタラクティブな機能を有する画面を表  
示するグラフィクスプレーンとがそれぞれ独立して設けられ、これら  
のプレーンが合成されて表示されるようにしているため、動画プレー  
ンに動画を表示させ、その動画を背景としてグラフィクスプレーンに  
メニュー画面などを表示させることができるという効果がある。

20 この発明によれば、グラフィクスプレーンに表示するための画像デ  
ータを格納するためのバッファが設けられているため、グラフィクス  
プレーンの表示を同一の画像データを繰り返し用いて表示することが  
できる。これにより、メニュー画面などをより高い自由度で構成する  
ことができるという効果がある。

25 この発明によれば、メニュー画面などに表示するボタンを 3 状態に  
分類し、それぞれに対して画像データを用意し、ユーザの入力などに

応じて切り替えることにより、従来のDVDビデオでは実現できなかった多彩な見栄えのメニューを実現できるという効果がある。

この発明では、グラフィクスプレーンに表示されるグラフィクスの表示制御を、表示制御命令を用いて記述しているため、インタラクティブな機能をグラフィクスプレーンに表示される画面を用いて実現することができるという効果がある。字幕やボタンのスクロール表示、移動表示といった簡単なアニメーションや、ユーザ入力に応じて画像の内容が変化する表現力のあるボタンを実現することができるという効果がある。

10 この発明によれば、記録済み大容量ディスクにおいて、動画を表示する動画プレーンと、字幕を表示する字幕プレーンと、メニュー画面などインタラクティブな機能を有する画面を表示するグラフィクスプレーンとがそれぞれ独立して設けられ、これらのプレーンが合成されて表示されるようにしている。そして、字幕プレーンとグラフィクス  
15 プレーンに表示するオブジェクトの形式として共通のグラフィクスオブジェクトを定義し、デコーダモデル、表示制御命令と動作モデルを定義した。これにより、動画に同期して字幕やボタンを表示することが可能となる効果がある。

この発明では、グラフィクスオブジェクトのデコーダモデルを定義  
20 し、実装のためにはオブジェクトバッファからプレーンバッファへの転送レートを制限する方法を示した。そして、同一のオブジェクトであっても、オブジェクトのプレーン上の位置、さらにオブジェクトの変形・移動などによって変化するプレーンバッファへの書き換えデータ量について、ウィンドウと称する矩形の更新領域を定義した。これ  
25 により、最小更新時間間隔を見積もることが可能となり、実装の実現性と再生互換性を高めることができるという効果がある。

この発明では、グラフィクスオブジェクトに音声データを含め、グラフィクスオブジェクトに含まれるボタン画像の表示と同時に音声データを再生するデコーダモデルを定義した。そのため、グラフィクスオブジェクトの表示に同期した音声データの再生を容易に行うことができるという効果がある。

この発明では、グラフィクスオブジェクトの表示制御命令に対して、音声データの再生を指示する命令を定義した。また、グラフィクスオブジェクトの表示制御命令に対して、グラフィクスオブジェクトに含まれる画像データに対して音声データを割り当てることができるようにしたため、任意の時刻での効果音などの音声データの再生や、効果音を伴う字幕やボタンの実現などが可能となる効果がある。



## 請 求 の 範 囲

1. コンテンツデータを再生する再生装置において、

少なくともプログラムコードおよびユーザに対して操作を促す操作画面を構成する画像データを含む非リアルタイムストリームと、少なくとも動画データおよび字幕データを含むリアルタイムストリームと、  
5 が入力される入力手段と、

上記入力手段により入力された上記プログラムコードを格納するコード格納手段と、

上記入力手段により入力された上記画像データを格納する画像データ格納手段と、  
10

上記入力手段により入力された上記動画データをデコードした復号動画データと、上記入力手段により入力された上記字幕データをデコードした復号字幕データとを合成する第1の合成手段と、

上記コード格納手段に格納された上記プログラムコードに基づき、  
15 上記画像データ格納手段に格納された上記画像データをデコードした復号画像データと、上記第1の合成手段により合成された動画字幕合成データとを合成する第2の合成手段と  
を有することを特徴とする再生装置。

2. 請求の範囲1に記載の再生装置において、

20 上記画像データの上記デコードは、上記プログラムコードに基づきユーザからの入力に応じて行われることを特徴とする再生装置。

3. 請求の範囲1に記載の再生装置において、

上記第2の合成手段による上記合成は、上記プログラムコードに基づきユーザからの入力に応じて行われることを特徴とする再生装置。

25 4. 請求の範囲1に記載の再生装置において、

上記プログラムコードは、HTMLまたはXHTMLに基づき記述

されていることを特徴とする再生装置。

5. 請求の範囲 1 に記載の再生装置において、

上記プログラムコードは、E C M A スクリプトで記述されていることを特徴とする再生装置。

5 6. 請求の範囲 1 に記載の再生装置において、

上記復号動画データの解像度を変換する解像度変換手段をさらに有し、上記第 1 の合成手段は、上記解像度変換手段の出力と上記復号字幕データとを合成するようにしたことを特徴とする再生装置。

7. 請求の範囲 1 に記載の再生装置において、

10 上記第 1 の合成手段による上記合成の度合いは、上記復号字幕データに応じて制御されることを特徴とする再生装置。

8. 請求の範囲 1 に記載の再生装置において、

上記第 2 の合成手段による上記合成の度合いは、上記復号画像データに応じて制御されることを特徴とする再生装置。

15 9. 請求の範囲 1 に記載の再生装置において、

上記復号字幕データの色情報を上記画像字幕合成データに用いる色として Y C b C r 形式の色情報として出力させる第 1 の色情報変換手段と、

20 上記復号画像データの色情報を R G B 形式から Y C b C r 形式に変換する第 2 の色情報変換手段と

をさらに有し、

上記第 1 の合成手段は、上記第 1 の色情報変換手段で色情報が変換された上記復号字幕データと、上記復号動画データとを合成し、

25 上記第 2 の合成手段は、上記第 2 の色情報変換手段で色情報が変換された上記復号画像データと、上記動画字幕合成データとを合成するようにしたことを特徴とする再生装置。

- 1 0. 請求の範囲 1 に記載の再生装置において、  
上記リアルタイムストリームは、MPEG 2 TS のトランスポート  
10 パケットに格納されて上記記録媒体から再生されることを特徴とする  
再生装置。
- 5 1 1. 請求の範囲 1 に記載の再生装置において、  
上記リアルタイムストリームに上記画像データがさらに含まれるこ  
とを特徴とする再生装置。
- 1 2. 請求の範囲 1 に記載の再生装置において、  
上記入力手段により入力された 1 または複数の上記字幕データを格  
10 納する字幕データ格納手段をさらに有し、  
上記第 1 の合成手段は、上記字幕データ格納手段に格納された複数  
の上記字幕データをデコードした複数の上記復号字幕データと、上記  
復号動画データとを合成できるようにしたことを特徴とする再生装置  
。
- 15 1 3. 請求の範囲 1 に記載の再生装置において、  
上記画像データ格納手段は、1 または複数の上記画像データを格納  
するようにされ、  
上記第 2 の合成手段は、上記画像データ格納手段に格納された複数  
の上記画像データを復号した複数の上記復号画像データと、上記動画  
20 字幕合成データとを合成できるようにしたことを特徴とする再生装置  
。
- 1 4. 請求の範囲 1 に記載の再生装置において、  
上記画像データおよび上記字幕データは、上記動画データに対して  
同期的に表示制御され、共通のデータ構造に格納されることを特徴と  
25 する再生装置。
- 1 5. 請求の範囲 1 4 に記載の再生装置において、

上記データ構造は、少なくとも、表示制御命令と、該表示制御命令により表示制御される 1 または複数の上記画像データまたは上記字幕データとを有することを特徴とする再生装置。

16. 請求の範囲 15 に記載の再生装置において、

- 5    上記表示制御命令は、上記画像データまたは上記字幕データ自体を変更せずに、該画像データまたは該字幕データの表示属性を変更する命令を含むことを特徴とする再生装置。

17. 請求の範囲 16 に記載の再生装置において、

上記表示属性は、透明度であることを特徴とする再生装置。

- 10   18. 請求の範囲 16 に記載の再生装置において、

上記表示属性は、表示色であることを特徴とする再生装置。

19. 請求の範囲 15 に記載の再生装置において、

- 上記データ構造は、同時に実行される複数の上記表示制御命令をひとまとめにして表示制御命令群として格納し、それぞれ異なる時刻に  
15   実行される複数の上記表示制御命令群を格納できるようにしたことを特徴とする再生装置。

20. 請求の範囲 14 に記載の再生装置において、

- 上記データ構造に格納される上記画像データまたは上記字幕データは、上記画像データをデコードする際の時間管理情報または上記字幕  
20   データをデコードする際の時間管理情報に基づき有効期間が開始され、該データ構造に定義される有効期間終了時刻で有効期間が終了されることを特徴とする再生装置。

21. 請求の範囲 1 に記載の再生装置において、

- 上記画像データはボタンを表示するボタン画像データであって、上  
25   記ボタンの 3 種類の状態にそれぞれ対応する第 1、第 2 および第 3 のボタン画像データをユーザの上記操作に従い差し替えて表示するよう

にしたことを特徴とする再生装置。

22. 請求の範囲14に記載の再生装置において、

上記画像データはボタンを表示するボタン画像データであって、上記ボタンの3種類の状態にそれぞれ対応する第1、第2および第3の  
5 ボタン画像データを同一の上記データ構造に格納するようにしたことを特徴とする再生装置。

23. 請求の範囲15に記載の再生装置において、

上記表示制御命令は、音声データの再生を制御する音声データ再生制御命令をさらに含めるようにしたことを特徴とする再生装置。

10 24. 請求の範囲23に記載の再生装置において、

上記音声データ再生制御命令に基づき上記字幕データの表示に対して同期的に音声データの再生を制御するようにしたことを特徴とする再生装置。

25. 請求の範囲23に記載の再生装置において、

15 上記画像データはボタンを表示するボタン画像データであって、上記音声データ再生制御命令に基づき上記ボタンの3種類の状態にそれぞれ対応する第1、第2および第3のボタン画像データの表示に応じて上記音声データを再生するようにしたことを特徴とする再生装置。

26. 請求の範囲23に記載の再生装置において、

20 上記音声データ再生制御命令により、上記画像データまたは上記字幕データに対して上記音声データを割り当てることができるようにしたことを特徴とする再生装置。

27. 請求の範囲1に記載の再生装置において、

上記入力手段に対して、効果音を再生する音声データがさらに入力  
25 されることを特徴とする再生装置。

28. 請求の範囲27に記載の再生装置において、

上記音声データは、上記リアルタイムストリームに多重化されて入力されることを特徴とする再生装置。

29. 請求の範囲28に記載の再生装置において、

上記リアルタイムストリームは画像データがさらに多重化され、該  
5 リアルタイムストリームに含まれ多重化される画像データおよび上記音声データは、1のデータ構造にまとめて格納されて入力されることを特徴とする記録媒体。

30. 請求の範囲29に記載の再生装置において、

上記データ構造には、上記リアルタイムストリームに多重化される  
10 画像データの表示制御および上記音声データの再生制御を行う制御命令がさらに格納されていることを特徴とする再生装置。

31. 請求の範囲29に記載の再生装置において、

上記データ構造には、上記リアルタイムストリームに多重化される  
15 画像データと対応する上記音声データが一組とされて格納されていることを特徴とする再生装置。

32. 請求の範囲31に記載の再生装置において、

上記リアルタイムストリームに多重化される画像データの表示に応じて該画像データと対応する上記音声データが再生されるようにしたことを特徴とする再生装置。

20 33. 請求の範囲32に記載の再生装置において、

上記リアルタイムストリームに多重化される画像データは、ボタンを表示するボタン画像データであって、上記データ構造には、上記ボタンの状態に対応した上記ボタン画像データと上記音声データとが一組とされて格納されていることを特徴とする再生装置。

25 34. 請求の範囲33に記載の再生装置において、

上記ボタンは複数の状態を有し、該複数の状態に複数のボタン画像

データがそれぞれ対応付けられると共に、該複数のボタン画像データの少なくとも1つに音声データが対応付けられて一組とされて、該複数のボタン画像データと該音声データとが同一の上記データ構造に格納されていることを特徴とする再生装置。

5 35. 請求の範囲34に記載の再生装置において、

上記ボタンの上記状態の変化に伴う上記ボタン画像データの表示に応じて該ボタン画像データと一組をなす上記音声データを再生するようにしたことを特徴とする再生装置。

36. 請求の範囲35に記載の再生装置において、

10 上記データ構造は、同時に実行される複数の上記制御命令をひとまとめにして制御命令群として格納し、それぞれ異なる時刻に実行される複数の上記制御命令群を格納できるようにされたことを特徴とする再生装置。

37. 請求の範囲27に記載の再生装置において、

15 上記音声データは、ファイルに格納されて上記非リアルタイムストリームとして上記入力手段から入力されることを特徴とする再生装置。

38. 請求の範囲37に記載の再生装置において、

20 上記リアルタイムストリームに多重化される画像データと、上記音声データを示す情報とが、1のデータ構造にまとめて格納されて入力されるようにしたことを特徴とする再生装置。

39. 請求の範囲38に記載の再生装置において、

25 上記データ構造は、上記リアルタイムストリームに多重化される画像データと、該画像データと対応する上記音声データを示す情報とが一組とされて格納され、該画像データの表示に応じて該画像データに対応する上記音声データが再生されるようにしたことを特徴とする再

生装置。

40. 請求の範囲39に記載の再生装置において、

上記リアルタイムストリームに多重化される画像データは、ボタン  
を表示するボタン画像データであって、上記データ構造には、上記ボ  
5 タンの状態に対応した上記ボタン画像データと、該ボタン画像データ  
と対応する上記音声データを示す情報とが一組とされて格納されてい  
ることを特徴とする再生装置。

41. 請求の範囲40に記載の再生装置において、

上記ボタンは複数の状態を有し、該複数の状態に複数のボタン画像  
10 データがそれぞれ対応付けられると共に、該複数のボタン画像データ  
の少なくとも1つに音声データが対応付けられて一組とされ、該複数  
のボタン画像データと該音声データを示す情報とが同一の上記データ  
構造に格納されていることを特徴とする再生装置。

42. 請求の範囲41に記載の再生装置において、

15 上記データ構造は、上記ボタン画像データの表示に応じて該ボタン  
画像データと対になる上記音声データを再生するようにした制御命令  
がさらに格納され、上記制御命令に基づき上記音声データを再生する  
ようにしたことを特徴とする再生装置。

43. 請求の範囲42に記載の再生装置において、

20 上記データ構造は、同時に実行される複数の上記制御命令をひとま  
とめにして制御命令群として格納し、それぞれ異なる時刻に実行され  
る複数の上記制御命令群を格納できるようにされていることを特徴と  
する再生装置。

44. 請求の範囲1に記載の再生装置において、

25 上記画像データをデコードする際のレートの上限を規定するように  
したことを特徴とする再生装置。



- 4 5. 請求の範囲 1 に記載の再生装置において、  
上記復号画像データをプレーンバッファに転送する際の転送レート  
の上限を規定するようにしたことを特徴とする再生装置。
- 4 6. 請求の範囲 4 5 に記載の再生装置において、
- 5 1 または複数の上記復号画像データを完全に含みむ上記プレーンバ  
ッファ上の矩形領域を定義し、画像データの描画、更新および消去の  
際に上記矩形領域を書き換えることで上記転送レートの上限を規定す  
るようにしたことを特徴とする再生装置。
- 4 7. 請求の範囲 1 に記載の再生装置において、
- 10 上記非リアルタイムストリームは、円盤状記録媒体から再生されて  
上記入力手段に入力されることを特徴とする再生装置。
- 4 8. 請求の範囲 1 に記載の再生装置において、  
上記非リアルタイムストリームは、ネットワークから取得されて上  
記入力手段に入力されることを特徴とする再生装置。
- 15 4 9. 請求の範囲 1 に記載の再生装置において、  
上記リアルタイムストリームは、円盤状記録媒体から再生されて上  
記入力手段に入力されることを特徴とする再生装置。
- 5 0. 請求の範囲 1 に記載の再生装置において、  
上記リアルタイムストリームは、ネットワークから取得されて上記  
20 入力手段に入力されることを特徴とする再生装置。
- 5 1. コンテンツデータを再生する再生方法において、  
少なくともプログラムコードおよびユーザに対して操作を促す操作  
画面を構成する画像データを含む非リアルタイムストリームと、少な  
くとも動画データおよび字幕データを含むリアルタイムストリームと  
25 を入力する入力のステップと、  
上記入力のステップにより入力された上記プログラムコードをコー

ド格納手段に格納するステップと、

上記入力のステップにより入力された上記画像データを画像データ格納手段に格納するステップと、

- 上記入力のステップにより入力された上記動画データをデコードした復号動画データと、上記入力のステップにより入力された上記字幕データをデコードした復号字幕データとを合成する第1の合成のステップと、

- 上記コード格納手段に格納された上記プログラムコードに基づき、上記画像データ格納手段に格納された上記画像データをデコードした復号画像データと、上記第1の合成のステップにより合成された動画字幕合成データとを合成する第2の合成のステップとを有することを特徴とする再生方法。

52. コンテンツデータを再生する再生方法をコンピュータ装置に実行させる再生プログラムにおいて、

- 15 上記再生方法は、

少なくともプログラムコードおよびユーザに対して操作を促す操作画面を構成する画像データを含む非リアルタイムストリームと、少なくとも動画データおよび字幕データを含むリアルタイムストリームとを入力する入力のステップと、

- 20 上記入力のステップにより入力された上記プログラムコードをコード格納手段に格納するステップと、

上記入力のステップにより入力された上記画像データを画像データ格納手段に格納するステップと、

- 上記入力のステップにより入力された上記動画データをデコードした復号動画データと、上記入力のステップにより入力された上記字幕データをデコードした復号字幕データとを合成する第1の合成のステ

ップと、

上記コード格納手段に格納された上記プログラムコードに基づき、  
上記画像データ格納手段に格納された上記画像データをデコードした  
復号画像データと、上記第 1 の合成のステップにより合成された動画

- 5 字幕合成データとを合成する第 2 の合成のステップと  
を有することを特徴とする再生プログラム。

5 3. コンテンツデータを再生する再生方法をコンピュータ装置に実  
行させる再生プログラムが記録された記録媒体において、

上記再生方法は、

- 10 少なくともプログラムコードおよびユーザに対して操作を促す操作  
画面を構成する画像データを含む非リアルタイムストリームと、少な  
くとも動画データおよび字幕データを含むリアルタイムストリームと  
を入力する入力のステップと、

- 上記入力のステップにより入力された上記プログラムコードをコー  
15 ド格納手段に格納するステップと、

上記入力のステップにより入力された上記画像データを画像データ  
格納手段に格納するステップと、

- 上記入力のステップにより入力された上記動画データをデコードし  
た復号動画データと、上記入力のステップにより入力された上記字幕  
20 データをデコードした復号字幕データとを合成する第 1 の合成のステ  
ップと、

- 上記コード格納手段に格納された上記プログラムコードに基づき、  
上記画像データ格納手段に格納された上記画像データをデコードした  
復号画像データと、上記第 1 の合成のステップにより合成された動画  
25 字幕合成データとを合成する第 2 の合成のステップと  
を有することを特徴とする記録媒体。

5 4. コンテンツデータが記録された円盤状の形状を有する記録媒体において、

少なくともプログラムコードおよびユーザに対して操作を促す操作画面を構成する画像データを含む非リアルタイムストリームと、少なくとも動画データおよび字幕データを含むリアルタイムストリームとが記録され、

上記プログラムコードに基づき、再生後に画像データ格納手段に格納された上記画像データをデコードした復号画像データと、再生された上記動画データをデコードした復号動画データおよび再生された上記字幕データをデコードした復号字幕データが合成された動画字幕合成データとが合成されることを特徴とする記録媒体。

5 5. 請求の範囲 5 4 に記載の記録媒体において、

上記画像データおよび上記字幕データは、上記動画データに対して同期的に表示制御され、共通のデータ構造に格納されて記録されることを特徴とする記録媒体。

5 6. 請求の範囲 5 5 に記載の記録媒体において、

上記データ構造は、少なくとも、表示制御命令と、該表示制御命令により表示制御される 1 または複数の上記画像データまたは上記字幕データとを有することを特徴とする記録媒体。

5 7. 請求の範囲 5 6 に記載の記録媒体において、

上記表示制御命令は、上記画像データまたは上記字幕データ自体を変更せずに、該画像データまたは該字幕データの表示属性を変更する命令を含むことを特徴とする記録媒体。

5 8. 請求の範囲 5 7 に記載の記録媒体において、

上記表示属性は、透明度であることを特徴とする記録媒体。

5 9. 請求の範囲 5 7 に記載の記録媒体において、

上記表示属性は、表示色であることを特徴とする記録媒体。

60. 請求の範囲56に記載の記録媒体において、

上記データ構造は、同時に実行される複数の上記表示制御命令をひとまとめにして表示制御命令群として格納し、それぞれ異なる時刻に  
5 実行される複数の上記表示制御命令群を格納できるようにして記録されることを特徴とする記録媒体。

61. 請求の範囲55に記載の記録媒体において、

上記画像データをデコードする際の時間管理情報または上記字幕データをデコードする際の時間管理情報に基づき有効期間が開始され、  
10 上記データ構造に定義される有効期間終了時刻で有効期間が終了されるようにした上記画像データまたは上記字幕データが上記データ構造に格納されて記録されることを特徴とする記録媒体。

62. 請求の範囲54に記載の記録媒体において、

上記画像データはボタンを表示するボタン画像データであって、上  
15 記ボタンの3種類の状態にそれぞれ対応する第1、第2および第3のボタン画像データをユーザの上記操作に従い差し替えて表示するようにしたことを特徴とする記録媒体。

63. 請求の範囲55に記載の記録媒体において、

上記画像データはボタンを表示するボタン画像データであって、上  
20 記ボタンの3種類の状態にそれぞれ対応する第1、第2および第3のボタン画像データを同一の上記データ構造に格納して記録することを特徴とする記録媒体。

64. 請求の範囲56に記載の記録媒体において、

上記表示制御命令は、音声データの再生を制御する音声データ再生  
25 制御命令をさらに含めて上記データ構造に格納され記録されることを特徴とする記録媒体。

65. 請求の範囲64に記載の記録媒体において、

上記音声データ再生制御命令に基づき上記字幕データの表示に対して同期的に音声データの再生を制御するようにした上記表示制御命令が上記データ構造に格納されて記録されることを特徴とする記録媒体

5。

66. 請求の範囲64に記載の記録媒体において、

上記画像データはボタンを表示するボタン画像データであって、上記音声データ再生制御命令に基づき上記ボタンの3種類の状態にそれぞれ対応する第1、第2および第3のボタン画像データの表示に応じて

10 上記音声データを再生するようにした上記表示制御命令が上記データ構造に格納されて記録されることを特徴とする記録媒体。

67. 請求の範囲64に記載の記録媒体において、

上記音声データ再生制御命令により上記画像データまたは上記字幕データに対して上記音声データを割り当てることができるようにした

15 上記表示制御命令が上記データ構造に格納されて記録されることを特徴とする記録媒体。

68. 請求の範囲54に記載の記録媒体において、

上記画像データがさらに上記リアルタイムストリームに多重化されて記録されることを特徴とする記録媒体。

20 69. 請求の範囲54に記載の記録媒体において、

効果音を再生する音声データがさらに記録されることを特徴とする記録媒体。

70. 請求の範囲69に記載の記録媒体において、

25 上記音声データは、上記リアルタイムストリームに多重化されて記録されることを特徴とする記録媒体。

71. 請求の範囲70に記載の記録媒体において、

上記リアルタイムストリームは画像データがさらに多重化され、該リアルタイムストリームに含まれる画像データおよび上記音声データを、1のデータ構造にまとめて格納して記録するようにしたことを特徴とする記録媒体。

5 72. 請求の範囲71に記載の記録媒体において、

上記データ構造には、上記リアルタイムストリームに多重化される画像データの表示制御および上記音声データの再生制御を行う制御命令がさらに格納されることを特徴とする記録媒体。

73. 請求の範囲71に記載の記録媒体において、

10 上記データ構造には、上記リアルタイムストリームに多重化される画像データに対応する上記音声データが一組とされて格納されることを特徴とする記録媒体。

74. 請求の範囲73に記載の記録媒体において、

15 上記リアルタイムストリームに多重化される画像データは、ボタンを表示するボタン画像データであって、上記データ構造には、上記ボタンの状態に対応した上記ボタン画像データと上記音声データとが一組とされて格納されることを特徴とする記録媒体。

75. 請求の範囲74に記載の記録媒体において、

20 上記ボタンは複数の状態を有し、該複数の状態に複数のボタン画像データがそれぞれ対応付けられると共に、該複数のボタン画像データの少なくとも1つに音声データが対応付けられ一組とされて、該複数のボタン画像データと該音声データとを同一の上記データ構造に格納することを特徴とする記録媒体。

76. 請求の範囲75に記載の記録媒体において、

25 上記ボタンの上記状態の変化に伴う上記ボタン画像データの表示に応じて該ボタン画像データと一組をなす上記音声データを再生するよ

うにされていることを特徴とする記録媒体。

77. 請求の範囲76に記載の記録媒体において、

上記データ構造は、同時に実行される複数の上記制御命令をひとまとめにして制御命令群として格納し、それぞれ異なる時刻に実行される複数の上記制御命令群を格納できるようにして記録されることを特徴とする記録媒体。

78. 請求の範囲69に記載の記録媒体において、

上記音声データは、ファイルに格納されて記録されることを特徴とする記録媒体。

10 79. 請求の範囲78に記載の記録媒体において、

上記リアルタイムストリームに多重化される画像データと、上記音声データを示す情報とを、1のデータ構造にまとめて格納して記録するようにしたことを特徴とする記録媒体。

80. 請求の範囲79に記載の記録媒体において、

15 上記データ構造には、上記リアルタイムストリームに多重化される画像データと、該画像データと対応する上記音声データを示す情報とが一組とされて格納されることを特徴とする記録媒体。

81. 請求の範囲80に記載の記録媒体において、

20 上記リアルタイムストリームに多重化される画像データは、ボタンを表示するボタン画像データであって、上記データ構造には、上記ボタンの状態に対応した上記ボタン画像データと、該ボタン画像データと対応する上記音声データを示す情報とが一組とされて格納されることを特徴とする記録媒体。

82. 請求の範囲81に記載の記録媒体において、

25 上記ボタンは複数の状態を有し、該複数の状態に複数のボタン画像データがそれぞれ対応付けられると共に、該複数のボタン画像データ



の少なくとも1つに音声データが対応付けられ一組とされて、該複数のボタン画像データと該音声データを示す情報とが同一の上記データ構造に格納されることを特徴とする記録媒体。

83. 請求の範囲82に記載の記録媒体において、

- 5   上記ボタン画像データの表示に応じて該ボタン画像データと対になる上記音声データを再生するようにした制御命令が上記データ構造にさらに格納されることを特徴とする記録媒体。

84. 請求の範囲83に記載の記録媒体において、

- 10   上記データ構造は、同時に実行される複数の上記制御命令をひとまとめにして制御命令群として格納し、それぞれ異なる時刻に実行される複数の上記制御命令群を格納できるようにして記録されることを特徴とする記録媒体。

85. 請求の範囲54に記載の記録媒体において、

- 15   予め規定された、上記画像データをデコードする際のレートの下限を満足するようにして上記画像データが記録されることを特徴とする記録媒体。

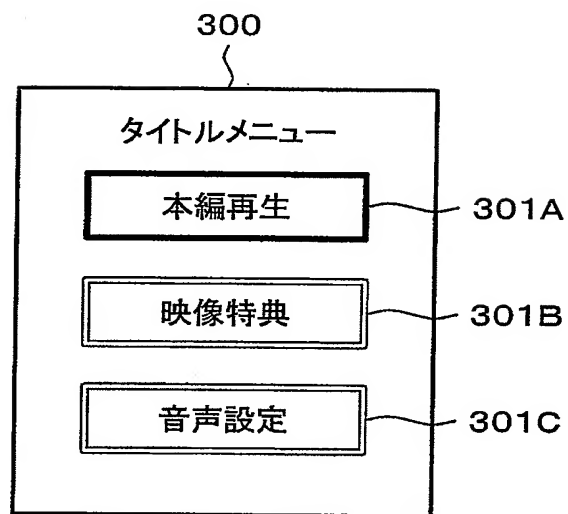
86. 請求の範囲54に記載の記録媒体において、

- 20   予め規定された、上記復号画像データをプレーンバッファに転送する際の転送レートの下限を満足するようにして上記画像データが記録されることを特徴とする記録媒体。

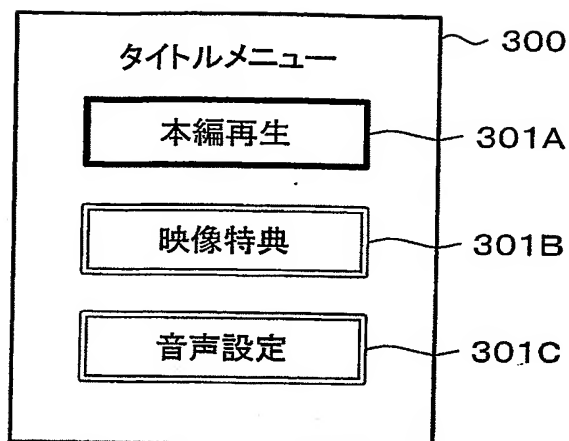
87. 請求の範囲86に記載の記録媒体において、

- 25   1または複数の上記復号画像データを完全に含みむ上記プレーンバッファ上の矩形領域を定義し、画像データの描画、更新および消去の際に上記矩形領域を書き換えることで上記転送レートの下限を満足するようにしたことを特徴とする記録媒体。

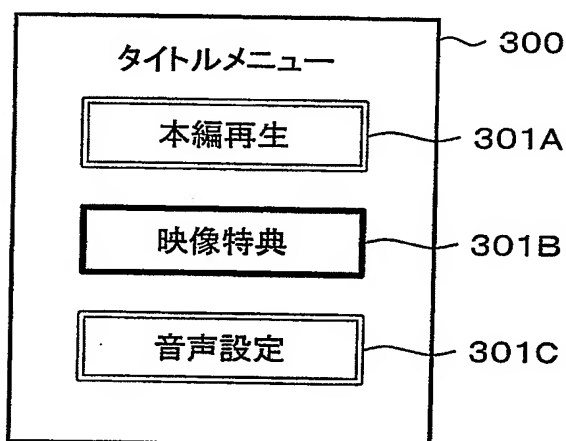
## 第1図



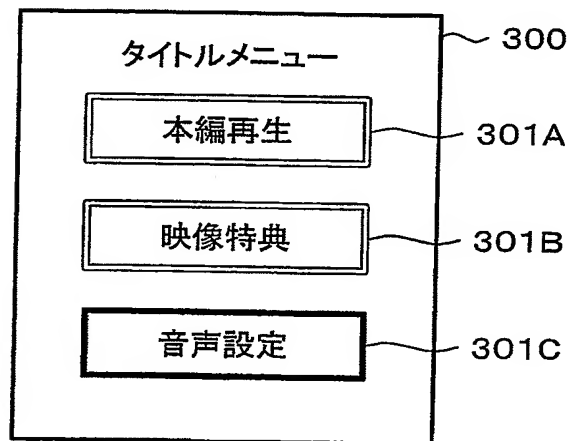
第2図A



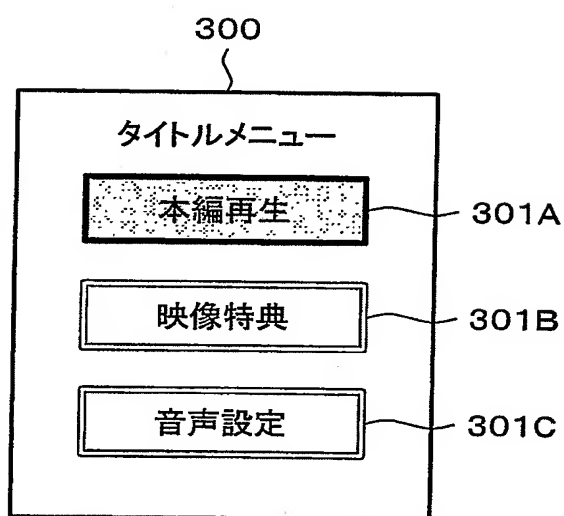
第2図B



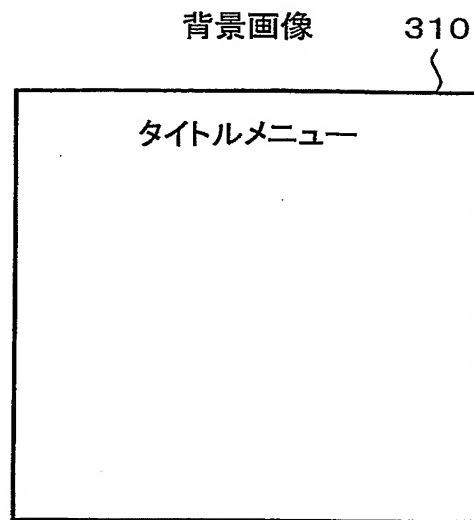
第2図C



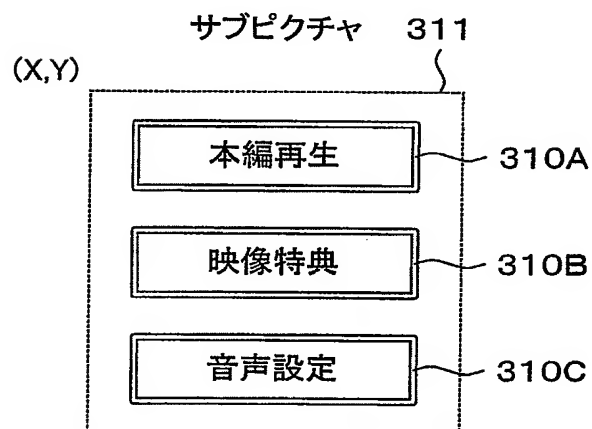
### 第3図



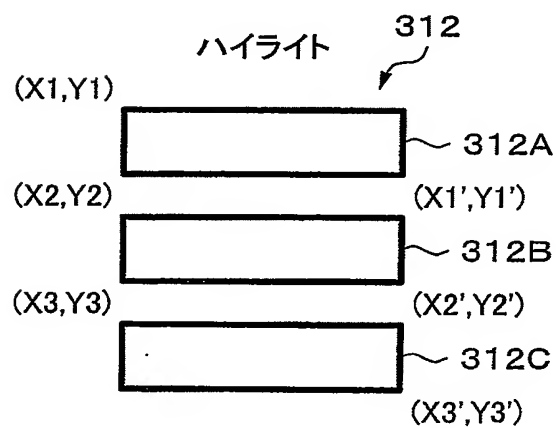
第4図A



第4図B



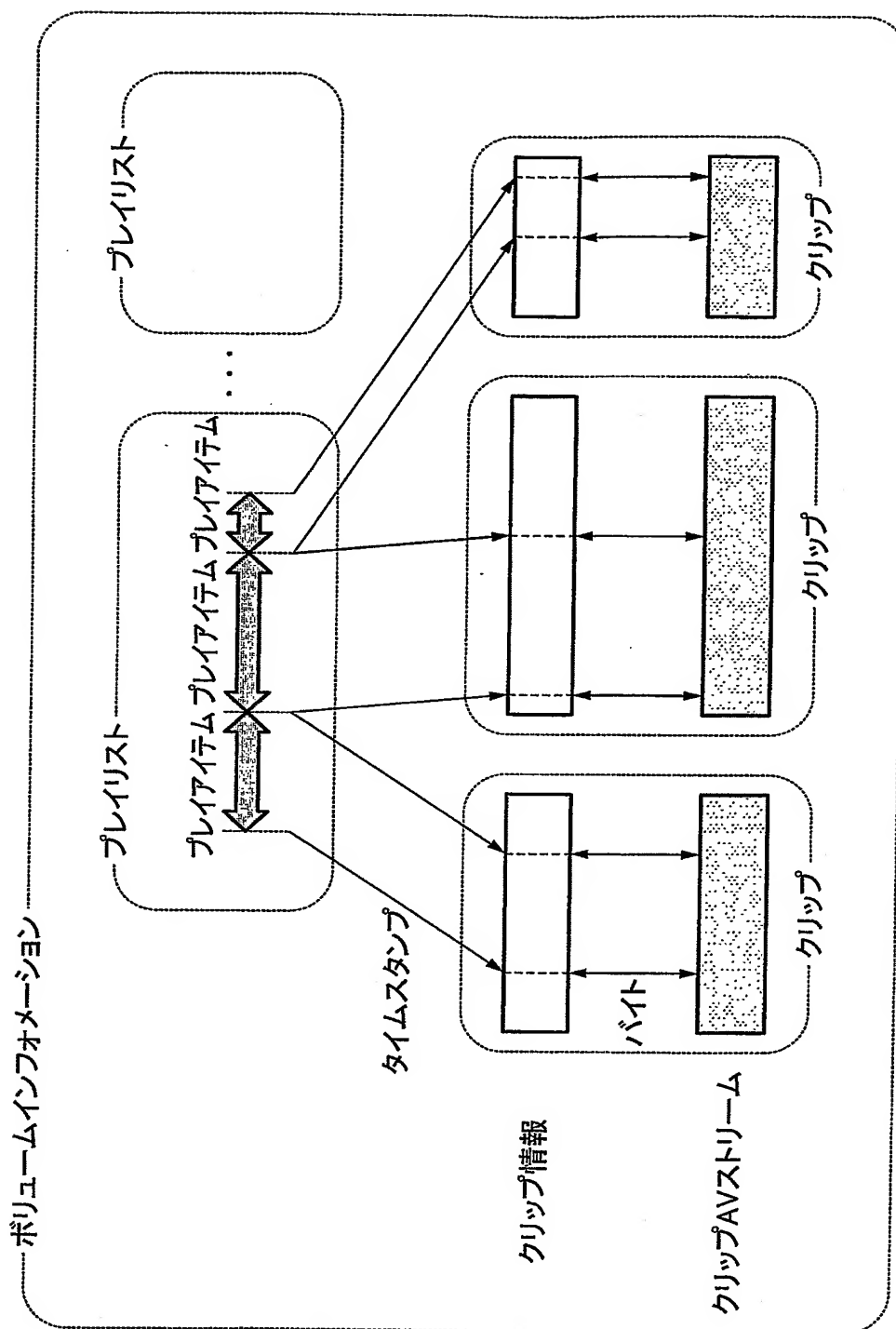
第4図C



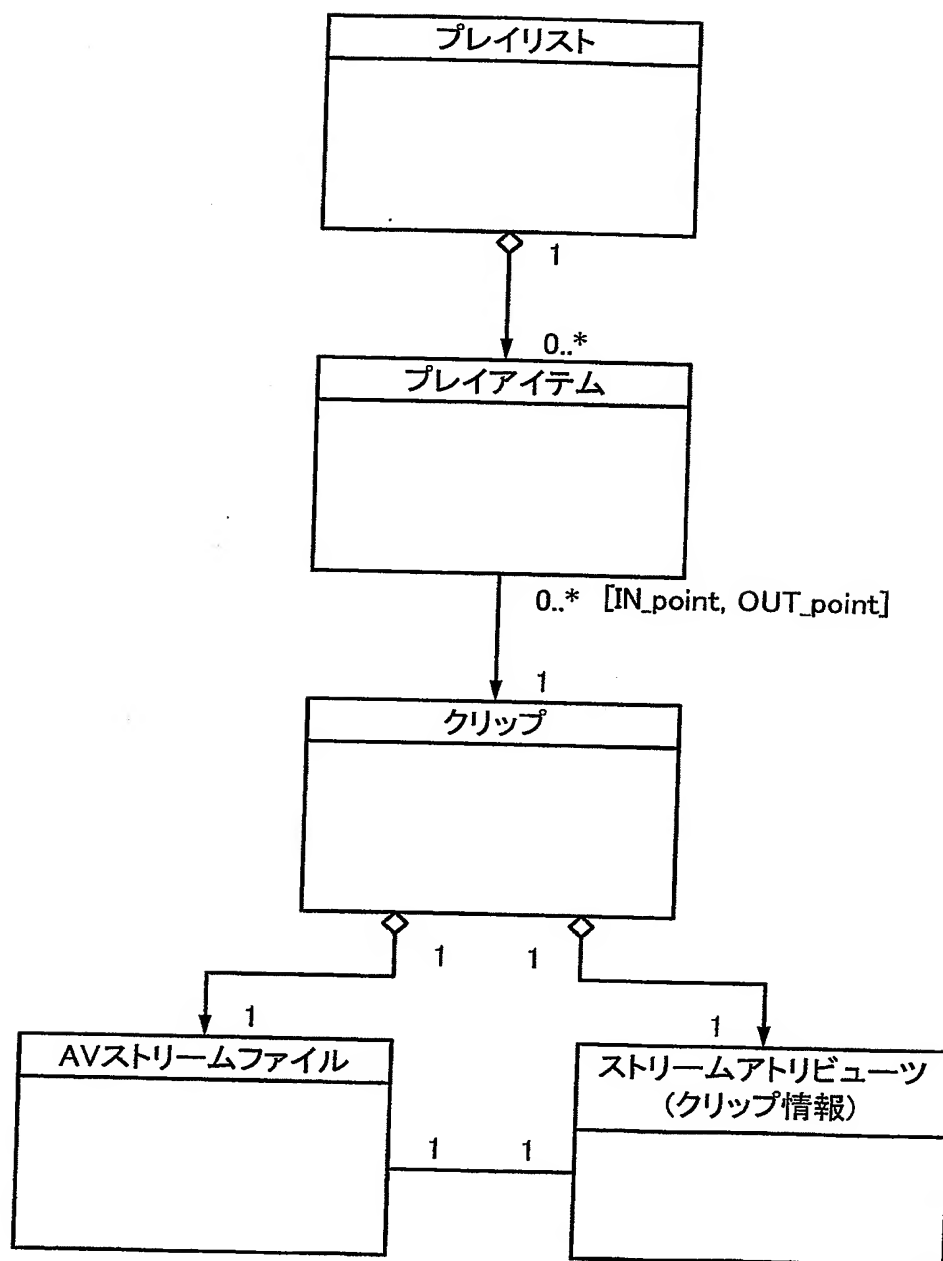
## 第5図

サブピクチャ	1ピクセルあたり2ビットの情報で表現された一枚の画像データ 色(A0,B0,C0,D0)
ハイライト	表示を開始する座標(X,Y) 選択状態の色(A1,B1,C1,D1) 実行状態の色(A2,B2,C2,D2) 色を変化させる領域の座標の組(=ボタンの数=3) (X1,Y1,X1',Y1') (X2,Y2,X2',Y2') (X3,Y3,X3',Y3')

## 第6図

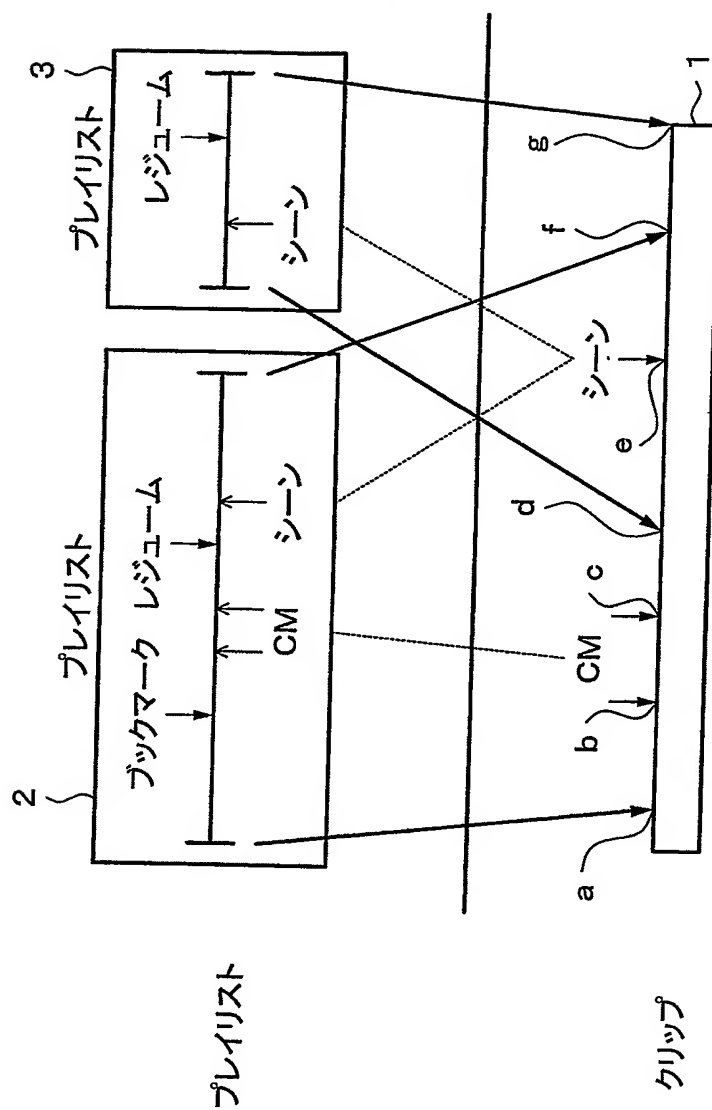


## 第7図

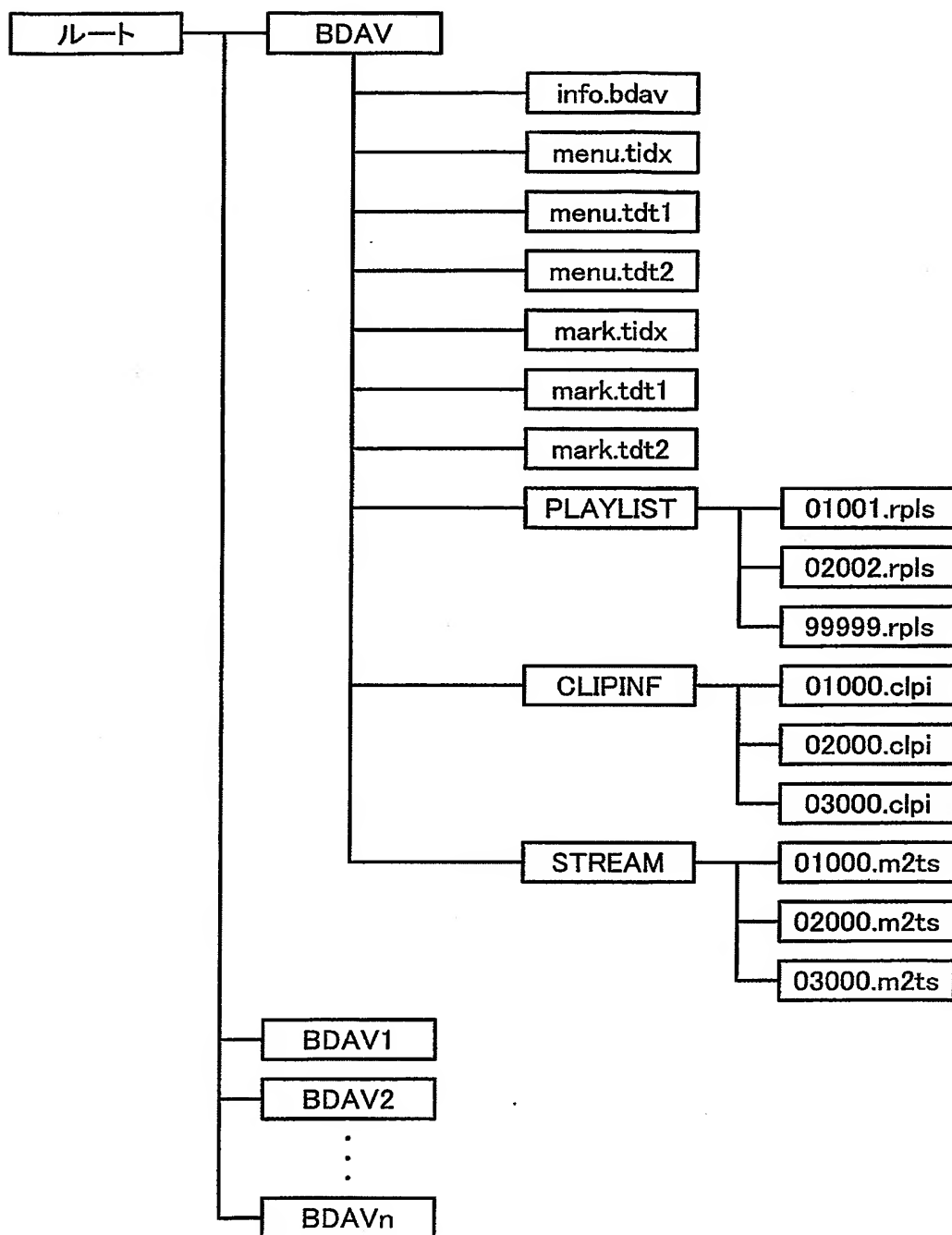




第8図



## 第9図



## 第 1 0 図

シンタクス	データ長(ビット)	ニーモニック
info.bdav{		
type_indicator	8*4	bslbf
version_number	8*4	bslbf
TableOfPlayLists_start_address	32	unimsbf
MakersPrivateData_start_address	32	unimsbf
reserved_for_future_use	192	bslbf
UIAppInfoBDav()		
for(i=0;i<N1;i++){		
padding_word	16	bslbf
}		
TableOfPlayLists()		
for(i=0;i<N2;i++){		
padding_word	16	bslbf
}		
MakersPrivateData()		
for(i=0;i<N3;i++){		
padding_word	16	bslbf
}		
}		

## 第 1 1 図

シンタクス	データ長(ビット)	ニーモニック
UIAppInfoBDAV(){		
length	32	unimsbf
reserved_for_future_use	16	bslbf
BDAV_character_set	8	bslbf
reserved_for_word_align	6	bslbf
BDAV_protect_flag	1	bslbf
resume_valid_flag	1	bslbf
PIN	8*4	bslbf
resume_PlayList_file_name	8*10	bslbf
ref_to_menu_thumbnail_index	16	unimsbf
BDAV_name_length	8	unimsbf
BDAV_name	8*255	bslbf
}		

## 第 1 2 図

シンタクス	データ長(ビット)	ニーモニック
TableOfPlayLists(){		
length	32	unimsbf
number_of_PlayLists	16	unimsbf
for(i=0;i<number_of_PlayLists;i++){		
PlayList_file_name	8*10	bslbf
}		
}		

## 第 1 3 図

シンタクス	データ長(ビット)	ニーモニック
xxxxx.rpls/yyyyy.vpls[		
type_indicator	8*4	bslbf
version_number	8*4	bslbf
PlayList_start_address	32	unimsbf
PlayListMark_start_address	32	unimsbf
MakersPrivateData_start_address	32	unimsbf
reserved_for_future_use	160	bslbf
UIAppInfoPlayList()		
for(i=0;i<N1;i++){		
padding_word	16	bslbf
}		
PlayList()		
for(i=0;i<N2;i++){		
padding_word	16	bslbf
}		
PlayListMark()		
for(i=0;i<N3;i++){		
padding_word	16	bslbf
}		
MakersPrivateData()		
for(i=0;i<N4;i++){		
padding_word	16	bslbf
}		
}		

## 第 1 4 図

シンタクス	データ長(ビット)	ニーモニック
UIAppInfoPlayList()[		
length	32	unimsbf
reserved_for_future_use	16	bslbf
PlayList_character_set	8	unimsbf
reserved_for_word_align	4	bslbf
playback_protect_flag	1	bslbf
write_protect_flag	1	bslbf
is_played_flag	1	bslbf
is_edited_flag	1	bslbf
time_zone	8	bslbf
reserved_for_word_align	8	bslbf
record_time_and_date	4*14	bslbf
PlayList_duration	4*6	bslbf
maker_ID	16	unimsbf
maker_model_code	16	unimsbf
channel_number	16	unimsbf
reserved_for_word_align	8	bslbf
channel_name_length	8	unimsbf
channel_name	8*20	bslbf
PlayList_name_length	8	unimsbf
PlayList_name	8*255	bslbf
PlayList_detail_length	16	unimsbf
PlayList_detail	8*1200	bslbf
]		

## 第15図

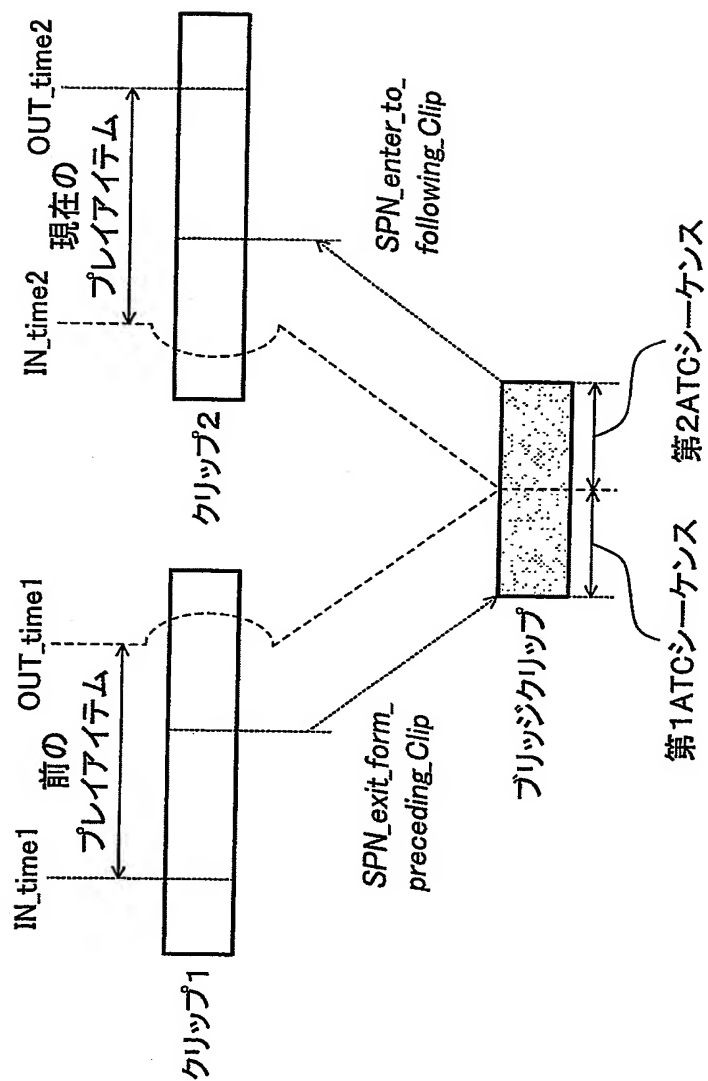
シンタックス	データ長(ビット)	ニーモニック
PlayList(){		
length	32	unimbsf
reserved_for_word_align	12	bslbf
PL_CPI_type	4	bslbf
number_of_PlayItems	16	unimbsf
if(<<Virtual-PlayList>&&PL_CPI_type==1){		
number_of_SubPlayItems	16	unimbsf
!else{		
reserved_for_word_align	16	bslbf
}		
for(PlayItem_id=0;PlayItem_id<number_of_PlayItems;PlayItem_id++){		
PlayItem()		
}		
if(<<Virtual-PlayList>&&CPI_type==1){		
for(i=0;i<number_of_SubPlayItems;i++){		
SubPlayItem()		
}		
}		
}		

## 第16図

シンタックス	データ長(ビット)	ニーモニック
PlayItem{		
length	16	unimbsf
Clip_Information_file_name	8*5	bslbf
Clip_codec_identifier	8*4	bslbf
reserved_for_future_use	6	bslbf
connection_condition	2	bslbf
if(CPL_type==1){		
ref_to_STC_id	8	unimbsf
}else{		
reserved_for_word_align	8	bslbf
}		
IN_time		
OUT_time	32	unimbsf
if(<Virtual-PlayList>&&connection_condition==3){	32	unimbsf
BridgeSequenceInfo{		
}		
}		



## 第17図



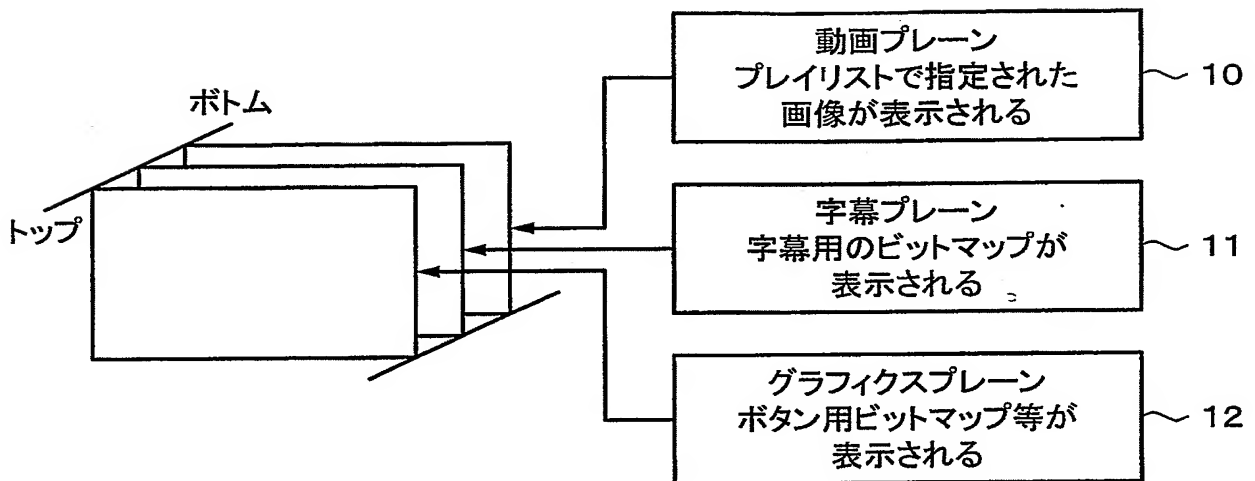
## 第18図

シンタックス	データ長(ビット)	ニーモニック
PlayListMark(){		
length		
number_of_PlayList_marks	32	unimsbf
for(i=0;i<number_of_PlayList_marks;i++){	16	unimsbf
mark_invalid_flag		
mark_type	1	unimsbf
mark_name_length	7	unimsbf
maker_ID	8	unimsbf
ref_to_PlayItem_id	16	unimsbf
mark_time_stamp	16	unimsbf
entry_ES_PID	32	unimsbf
if(mark_type==0x01  mark_type==0x02){	16	unimsbf
ref_to_menu_thumbnail_index		
}else{	16	unimsbf
ref_to_menu_thumbnail_index		
}	16	unimsbf
duration		
makers_information	32	unimsbf
mark_name	32	bslbf
}	8*24	bslbf
}		

## 第 19 図

シンタクス	データ長(ビット)	ニーモニック
zzzzz.clpi{		
type_indicator	8*4	bslbf
version_number	8*4	bslbf
SequenceInfo_start_address	32	unimsbf
ProgramInfo_start_address	32	unimsbf
CPI_start_address	32	unimsbf
ClipMark_start_address	32	unimsbf
MakersPrivateData_start_address	32	unimsbf
reserved_for_future_use	96	bslbf
ClipInfo()		
for(i=0;<N1;i++){		
padding_word	16	bslbf
}		
SequenceInfo()		
for(i=0;<N2;i++){		
padding_word	16	bslbf
}		
ProgramInfo()		
for(i=0;<N3;i++){		
padding_word	16	bslbf
}		
CPI()		
for(i=0;<N4;i++){		
padding_word	16	bslbf
}		
ClipMark()		
for(i=0;<N5;i++){		
padding_word	16	bslbf
}		
MakersPrivateData()		
for(i=0;<N6;i++){		
padding_word	16	bslbf
}		
}		

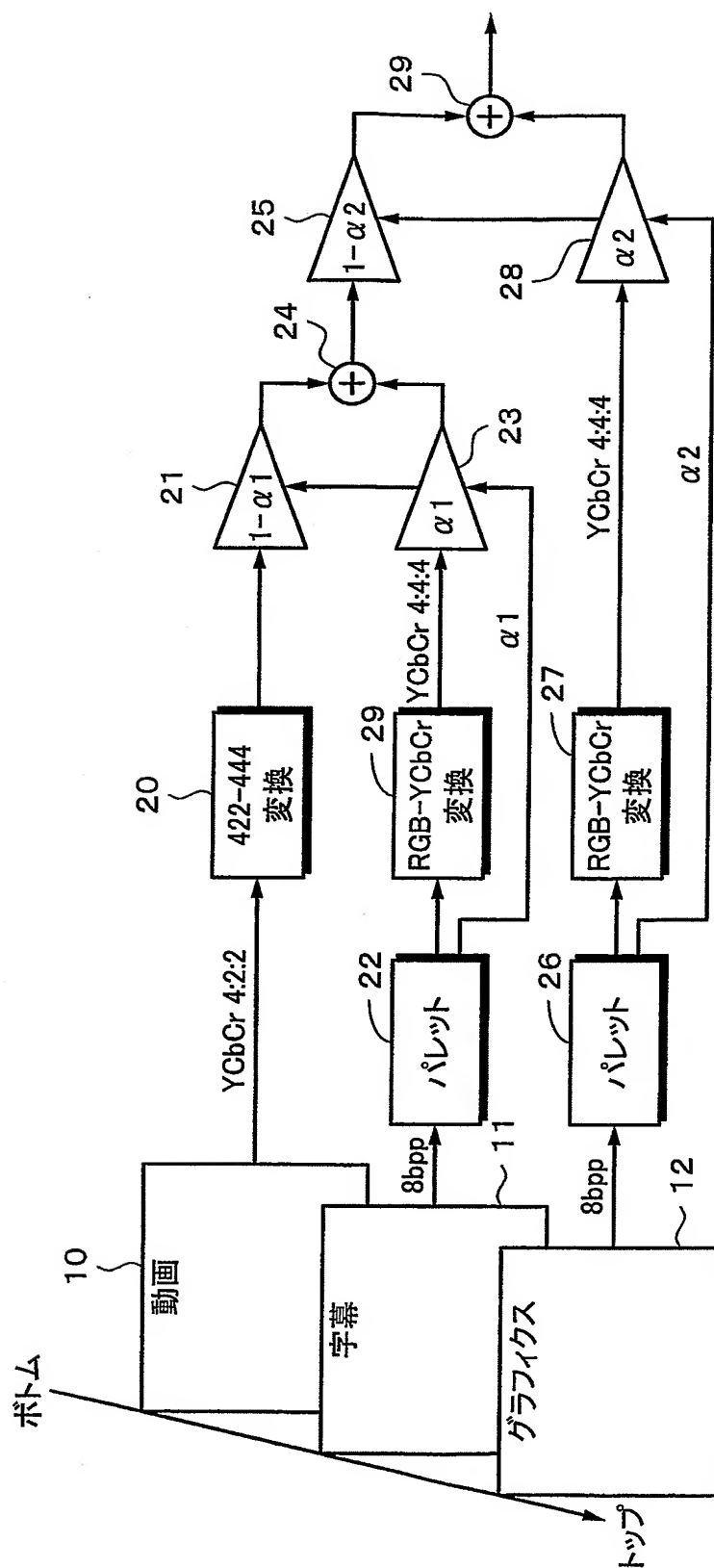
## 第 2 0 図



## 第 2 1 図

項目	規定内容
動画プレーン	1920x1080x16ビット, YCbCr(4:2:2),各8ビット
字幕プレーン	1920x1080x8ビット, 8ビットカラーマップアドレス(パレット) +256段階のアルファ ブレンディング
グラフィックスプレーン	1920x1080x8ビット, 8ビットカラーマップアドレス(パレット)+ 256段階のアルファ ブレンディング

## 第22図



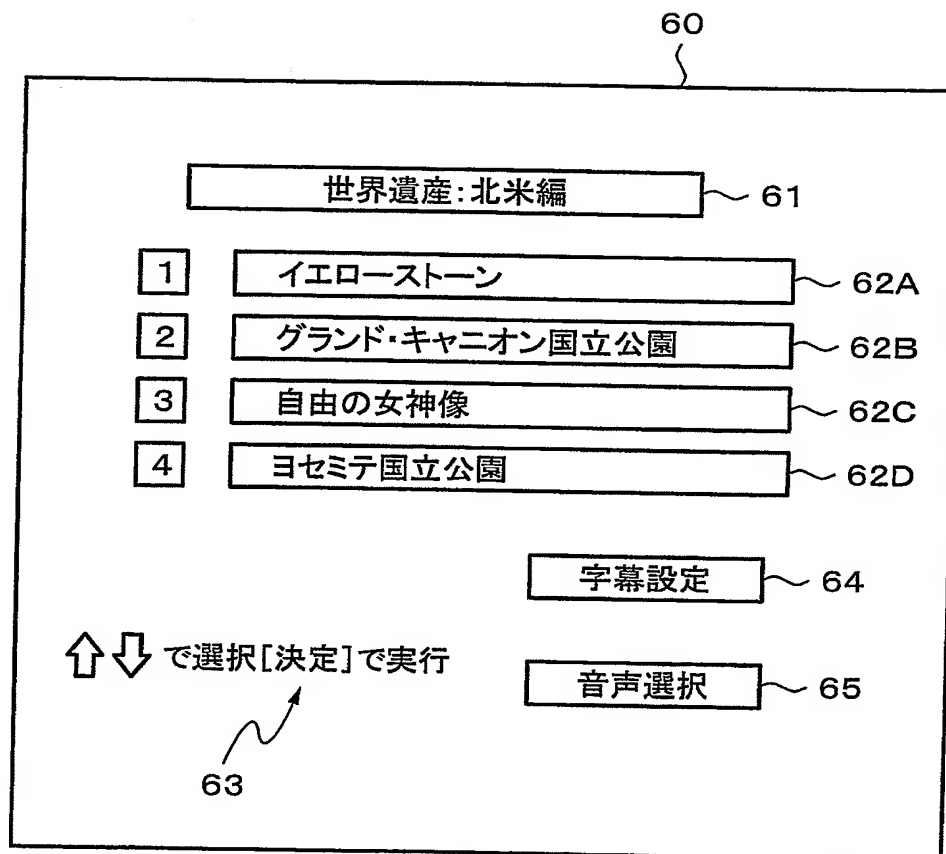
## 第 2 3 図

入力	入力アドレス 8ビット
出力	出力データ 8ビットx4、(R, G, B, $\alpha$ )出力

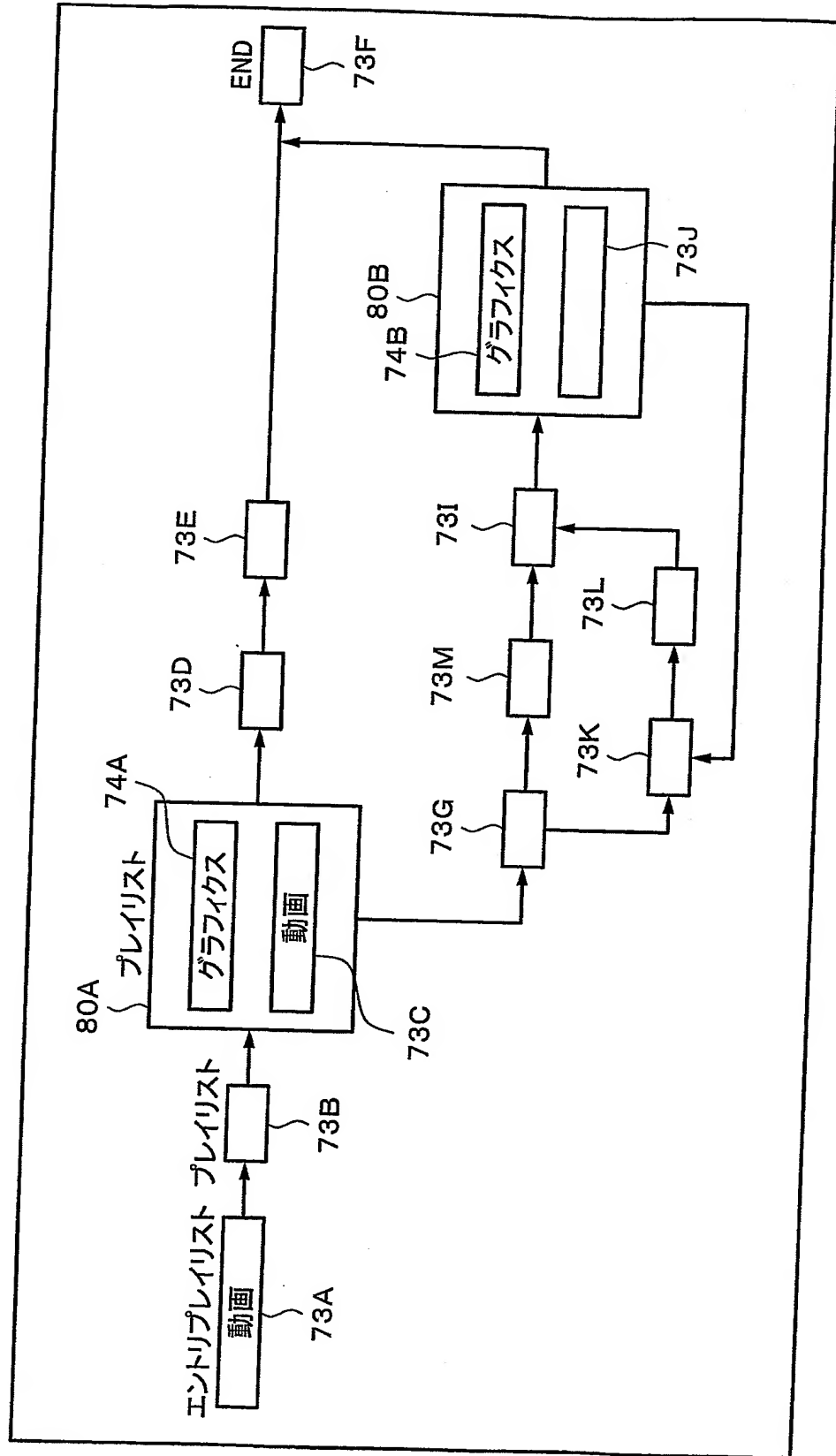
## 第 2 4 図

カラーインデックス値	三原色の値			不透明度
	R	G	B	$\alpha$
0x00	0	0	0	0
0x01	10	100	30	0.5
⋮	⋮	⋮	⋮	⋮
⋮	⋮	⋮	⋮	⋮
0xFF	200	255	100	0.8

## 第 2 5 図

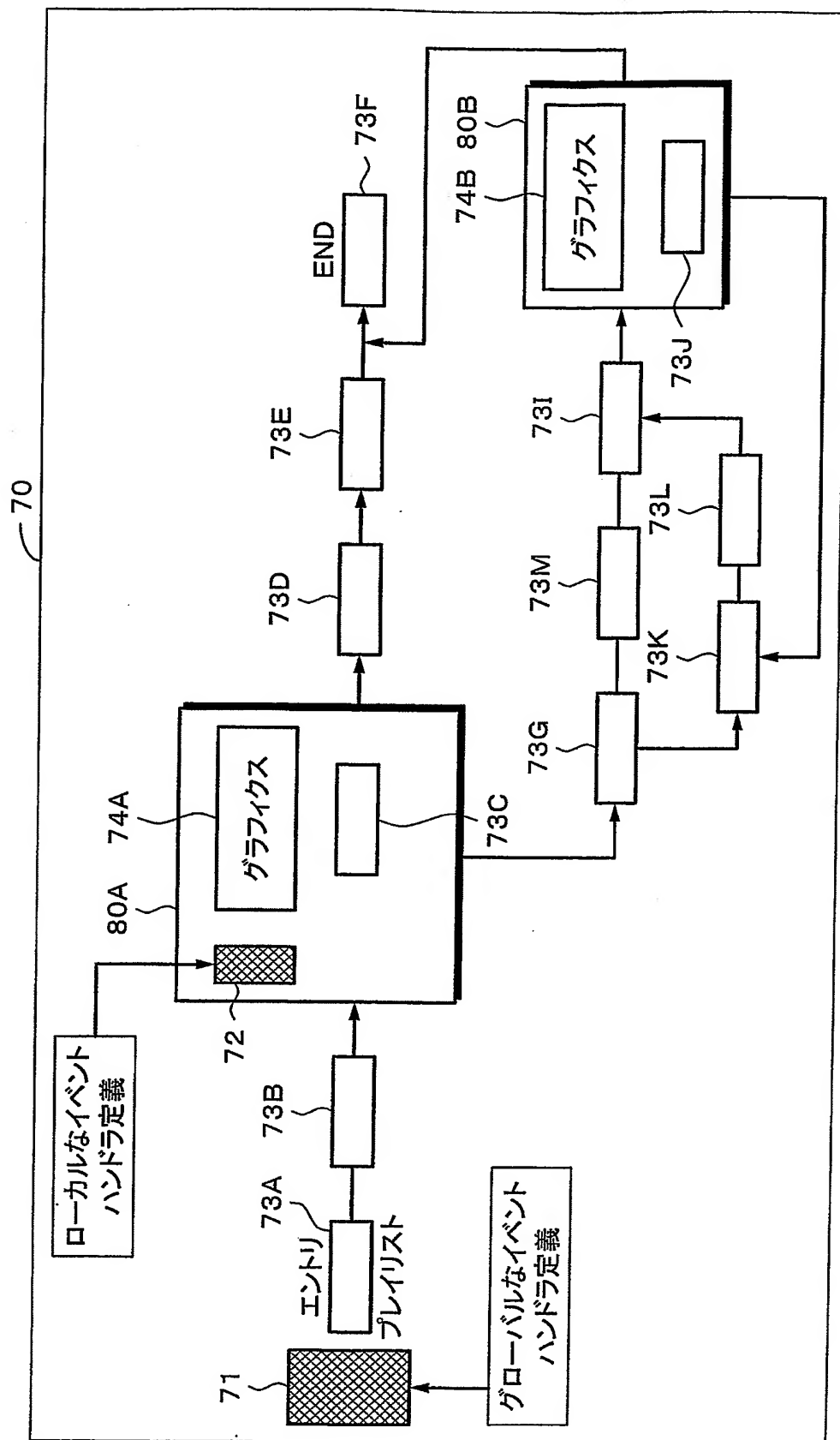


第26図

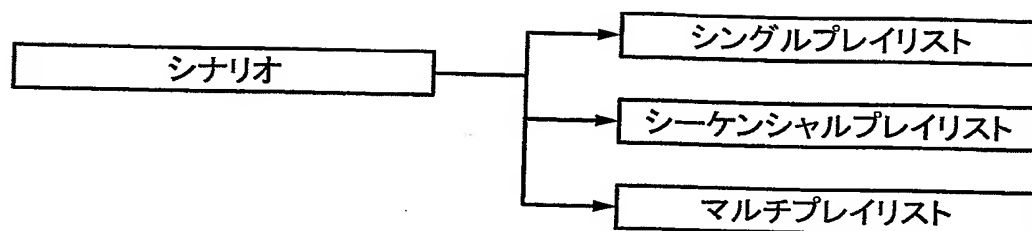




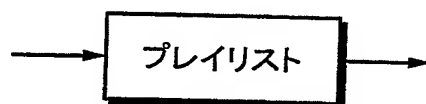
第 27 圖



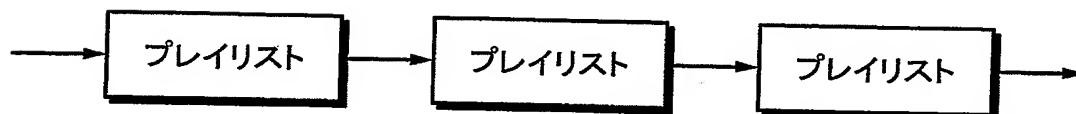
第 2 8 図



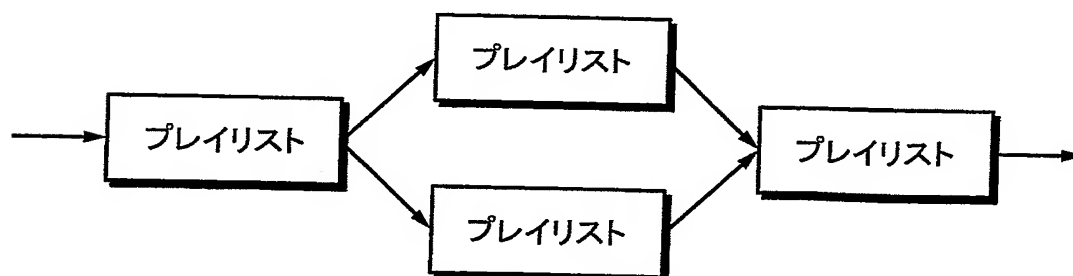
第 2 9 図 A



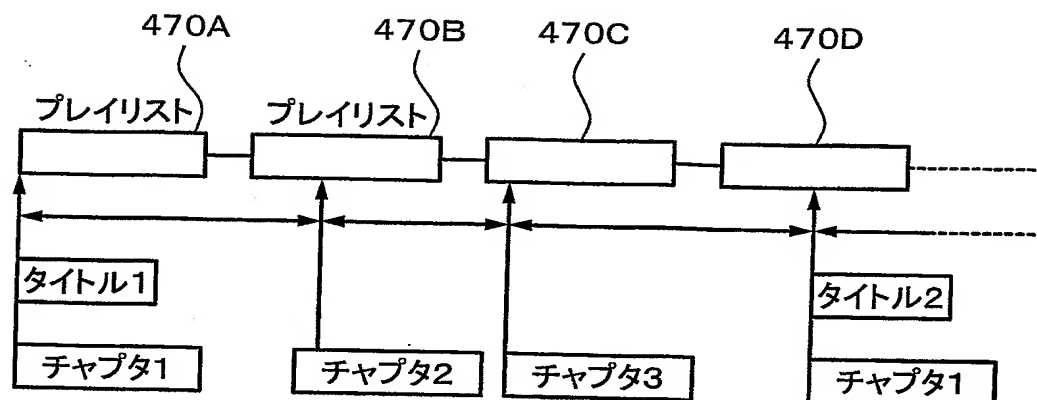
第 2 9 図 B



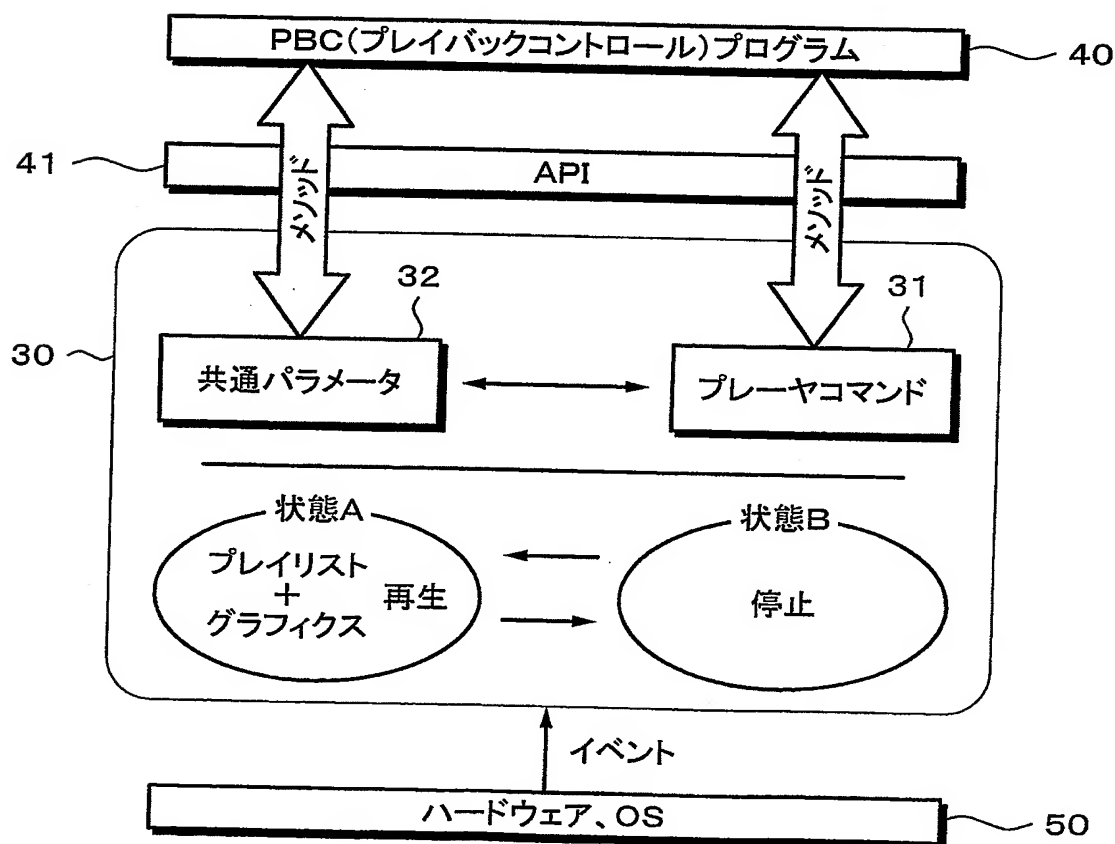
第 2 9 図 C



## 第30図



## 第31図



## 第32図A

第32図

第32図A
第32図B
第32図C

イベント名	説明
TimeFired	カウントダウンタイマーが0になった カウントアップタイマーが指定の値になった
PlayStopped	再生が停止した
PlayStilled	再生が一時停止した
StillReleased	一時停止が解除された
PlayPaused	キューにより一時停止された
PauseReleased	ポーズが解除された

## 第32図B

イベント名	説明
PlayStarted	再生が開始された
PlayRepeated	繰り返し再生時の先頭を検出した
SPDisplayStatusChanged	SPストリームの表示・非表示状態が変わった
SelectedAudioChanged	再生するオーディオストリームが変わった
VideoStopped	再生するビデオストリームが変わった
ScenarioStarted	シナリオの先頭を検出
ScenarioEnded	シナリオの終わりを検出
PlayListStarted	プレイリストの先頭を検出
PlayListEnded	プレイリストの終わりを検出
PlayItemStarted	プレイアイテムの先頭を検出
PlayItemEnded	プレイアイテムの終わりを検出

## 第32図C

イベント名	説明
MarkEncountered	再生中にマークを検出 グラフィックス画面を表示する際に利用 共通パラメータに、種類と番号が格納される
ButtonPressed	画面上のボタンが押された 共通パラメータに、押されたボタンのIDが格納される
ValidPeriodStarted	有効期間が開始した リンクの選択有効期間を設定する際に利用
ValidPeriodEnded	有効期間が終了した リンクの強制実行の時に利用
KeyPressed	リモコンのキーが押された どのキーが押されたかは、イベントハンドラ内のスイッチなどで識別する

第 3 3 図 A

第 3 3 図

第 3 3 図 A
第 3 3 図 B
第 3 3 図 C
第 3 3 図 D
第 3 3 図 E
第 3 3 図 F
第 3 3 図 G
第 3 3 図 H

メソッド	備考
再生開始位置指定	
LinkPlayList(playListNumber)	playListNumberで指定されたPlayListの再生を開始する。
LinkPlayItem(playListNumber,playItemNumber)	指定のPlayListの指定のPlayItemから再生を開始する。 playItemNumberはPlayItem_idであり、0から始まる。 PlayListの先頭から再生するのであれば、playItemNumberは0になる。

## 第33図B

メソッド	備考
再生開始位置指定	
Link(position)(object) position=("prev" "next" "top" "Parent" "tail") object=(Playlist   PlayItem   Chapter)	シナリオ内移動。 現在再生中の箇所から前後のPlaylist, PlayItem, Chapterに移動する。
Exit	シナリオの再生を停止する。標準レジスタの値は保持されない。
RSM	前回再生を停止した箇所から再生を再開する。 保存されているresume情報を呼び出し、レジスタにセットして再生を開始する。



## 第33図C

Player状態の取得	
getMenuDescriptionLanguage()	メニュー表示の際の言語を取得
getScenarioNumber()	再生中のシナリオ番号を得る。
getPlayListNumber()	再生中のプレイリスト番号を得る。
getChapterNumber()	再生中のチャプター番号を得る。
getPlayerSupport()	プレーヤのバージョン、有する機能を取得。

## 第33図D

Videoストリームに関するもの	
getVideoStreamAvailability()	指定のビデオストリームが含まれているかを得る。
setVideoStreamNumber()	デコードするビデオストリームを指定する。
getVideoStreamNumber()	選択中のビデオストリームの番号を得る。
getVideoStreamAttribute()	ビデオストリームの属性を得る。 (符号化方式、解像度、アスペクト比、4:3時のディスプレイモード、ClosedCaption)
setAngleNumber()	アングル番号を指定する。
getAngleNumber()	選択中のアングル番号を得る。
getMaxVideoStreams()	選択可能なビデオストリームの数を得る。 getVideoStreamAvailability()があれば十分か？

## 第33図E

	オーディオストリームに関するもの	
	getAudioStreamAvailability()	指定のオーディオストリームが含まれているかを得る。
	getAudioStreamLanguage()	指定のオーディオストリームの言語についての情報を得る。
	getAudioStreamStatus()	再生するオーディオストリームを指定する。
	setAudioStreamStatus()	再生しているオーディオストリームの番号を得る。
	getAudioStreamAttribute()	オーディオストリームの属性を得る。(符号化方式、ch数、Q、fs)

第33図F

サブピクチャストリーム	
getSPStreamAvailability()	指定のSPストリームが含まれているかを得る。
getSPStreamLanguage()	指定のSPストリームの言語を得る。
getSPDisplayStatus()	SPの表示状態(表示または非表示)を得る。
setSPDisplayStatus()	SPの表示状態(表示または非表示)を設定する。
getSpStreamAttribute()	SPの属性を得る。解像度別、4:3 またはワイド用など。

## 第33図G

レジスタリード/ライト	
clearReg()	全レジスタの初期化する。
setReg()	レジスタへの値のセット
getReg()	レジスタから値の読み出し

## 第33図H

タイマ	
sleep()	指定されたミリ秒間、処理を停止する。
setTimeout()	指定されたミリ秒後に関数や処理を実行する。
setInterval()	指定されたミリ秒ごとに処理を実行する。
clearTimer()	指定された登録タイマIDの処理を中止する。
pauseTimer()	指定した登録タイマIDのタイマを一時停止する。
resumeTimer()	指定した登録タイマIDのタイマを一時停止から再開する。
その他	
playSoundEffect(sound_id)	選択された効果音を再生。ボタンコマンドで利用できる。

## 第34図A

第34図

第34図A
第34図B
第34図C
第34図D
第34図E
第34図F
第34図G
第34図H
第34図I
第34図J

オブジェクト	メソッド	備考
bdp		
プレーヤー動作		
	playScenario(scenarioNumber,[scenarioTime])	scenarioNumberで指定されるシナリオを再生する。 ScenarioNumberは、実際にはシナリオ構成を記述した ファイル等を指すURIになる。 scenarioTimeはオプションであるが、 シナリオ内の経過時刻を指定する。
	playPlayList(playListNumber)	playListNumberで指定されたPlayListを再生する。
	playChapterMark(playListNumber,chapterNumber)	playListNumberで決まるPlayList中の chapterNumberで指定されるあるチャプターから再生する。

## 第34図B

	playPlayItem(playListNumber,playItemNumber)	playListNumberとplayItemNumberで決まる PlayItemから再生する。 playItemNumberはPlayItem_idであり、0から始まる。 PlayListの先頭から再生するのであれば、 playItemNumberは0になる。
	play(position)(object) position = ("prev" "next" "top" "goUp" "tail") object = (PlayList PlayItem Chapter)	シナリオ内移動。現在再生中の箇所から前後の PlayList,PlayItem等に移動する。
	stop()	シナリオの再生を停止する。 標準レジスタの値は保持されない。
	resume()	前回再生を停止した箇所から再生を再開する。
	playSoundEffect()	選択された効果音を再生



## 第34図C

プレーヤー状態	
	メニュー表示の際の言語を取得
getMenuDescriptionLanguage()	
	再生中のシナリオ番号を得る。
getScenarioNumber()	
	再生中のプレイリスト番号を得る。
getPlayListNumber()	
	再生中のチャプター番号を得る。
getChapterNumber()	
	プレイヤーのバージョン、有する機能を取得
getPlayerSupport()	

## 第34図D

ビデオストリーム		
	setVideoStreamNumber()	デコードするビデオストリームを指定する。
	getVideoStreamNumber()	選択中のビデオストリーム番号を得る。
	getVideoStreamStatus()	ビデオストリームの状態を得る。
	getVideoStreamAttr()	ビデオストリームの属性を得る。 (符号化方式、解像度、アスペクト比、 4:3時のディスプレイモード、クローズドキャプション)
	setAngleNumber()	アングル番号を指定する。
	getAngleNumber()	選択中のアングル番号を得る。
	getMaxVideoStreams()	ビデオストリームの最大数を得る。

## 第34図E

オーディオストリーム		
	getAudioStreamAvailability()	指定のオーディオストリームが含まれているかを得る。
	getAudioStreamLanguage()	指定のオーディオストリームの言語についての情報を得る。
	getAudioStreamStatus()	オーディオストリームの状態(再生または非再生)をセットする。
	setAudioStreamStatus()	オーディオストリームの状態(再生または非再生)をセットする。
	getAudioStreamAttribute()	オーディオストリームの属性を得る。(符号化方式、ch数、Q、fs)

## 第34図F

サブピクチャストリーム		
	getSPStreamAvailability()	指定のSPストリームが含まれているかを得る。
	getSPStreamLanguage()	指定のSPストリームの言語を得る。
	getSPDisplayStatus()	SPの表示状態を得る。
	setSPDisplayStatus()	SPの表示状態を設定する。SPの表示のON/OFF
	getSpStreamAttribute()	SPの属性を得る。4:3またはワイド用など。

## 第34図G

レジスタリード/ライト	
	clearReg()
	setReg()
	getReg()

プレイヤが内蔵しているメモリ領域(レジスタ)に対する操作。  
全レジスタを初期化する。

レジスタへの値のセット

レジスタから値の読み出し

## 第34図H

タイマ	
sleep()	指定されたミリ秒間、処理を停止する。
setTimeout()	指定されたミリ秒後に関数や処理を実行する。
setInterval()	指定されたミリ秒ごとに処理を実行する。
clearTimer()	指定された登録タイマIDの処理を中止する。
pauseTimer()	指定した登録タイマIDのタイマを一時停止する。
resumeTimer()	指定した登録タイマIDのタイマを一時停止から再開する。

## 第34図I

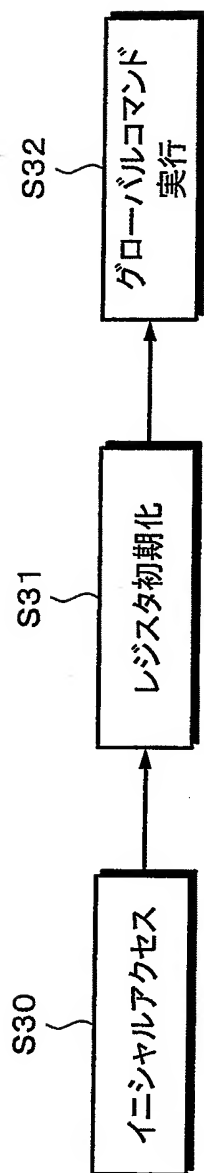
キー入力		
	getPressedKey()	入力されたキーの種類を取得
グラフィックス関連		
	loadGraphics(htmlfile, ID)	htmlfileで指定されるファイルを読み込み、グラフィックスプレーンに展開する。 ただし、表示はしない。この画像は、IDで参照される。
	showGraphics(ID)	IDで指定されるグラフィックス画像を表示する。 すでにloadGraphics()でプレーン上に展開されていることが必要。
	hideGraphics(ID)	IDで指定されるグラフィックス画像を消す。

## 第34図J

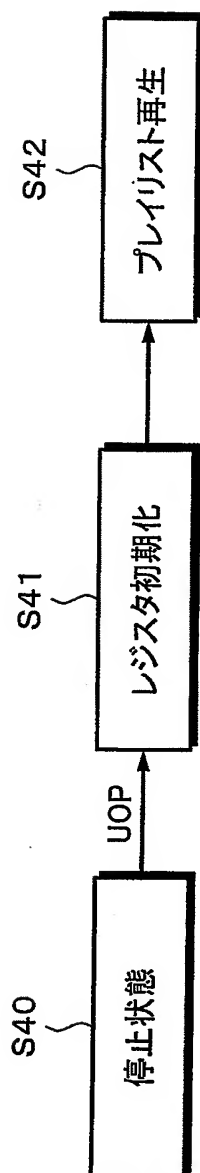
その他		
	random(input Number num)	1からnumまでの乱数を発生する。 ECMASクリプトのMathオブジェクトを使用せず、独自に定義する。
	catchEvent(eventname,eventhandler)	eventnameで指定されたイベントが発生した際、 eventhandlerで指定した関数を実行することを登録する。



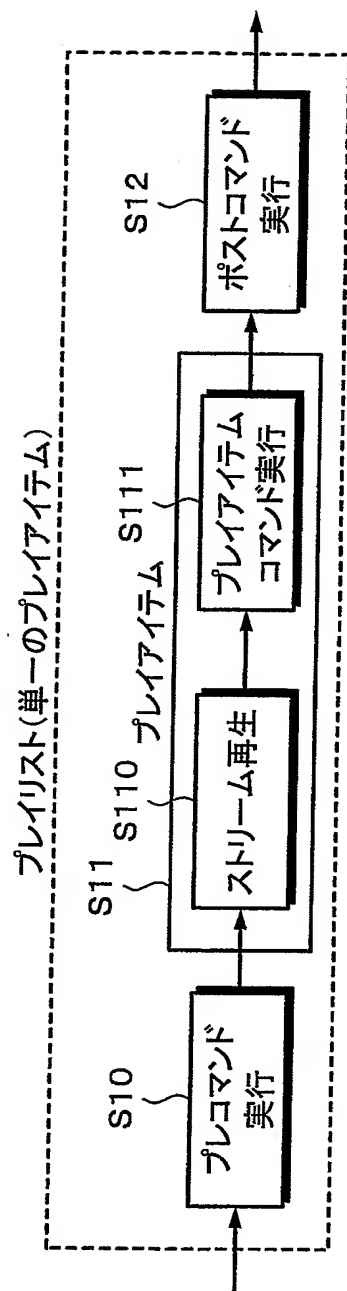
第35図A



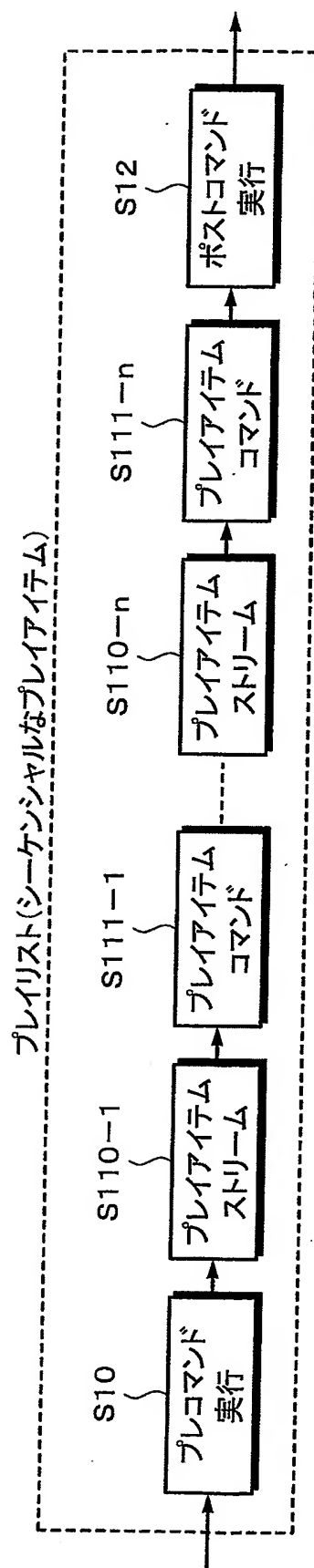
第35図B



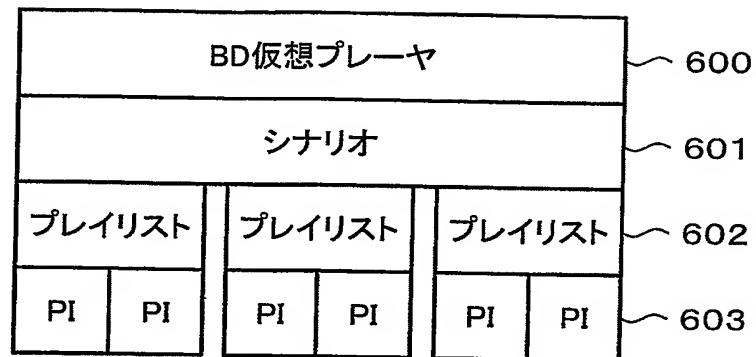
第36図A



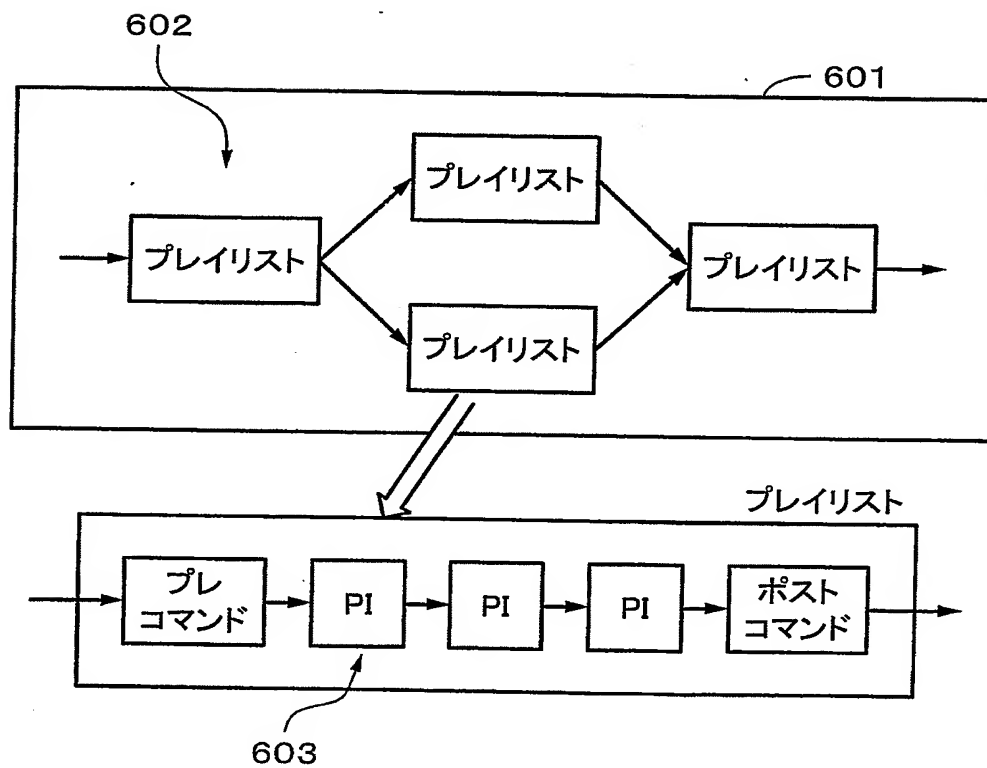
第36図B



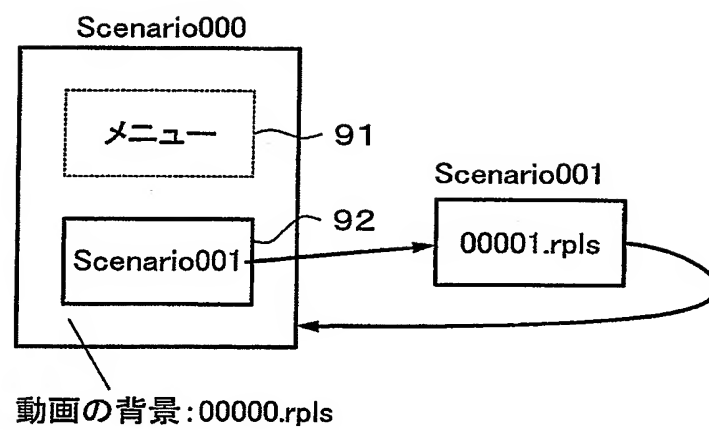
## 第 3 7 図 A



## 第 3 7 図 B



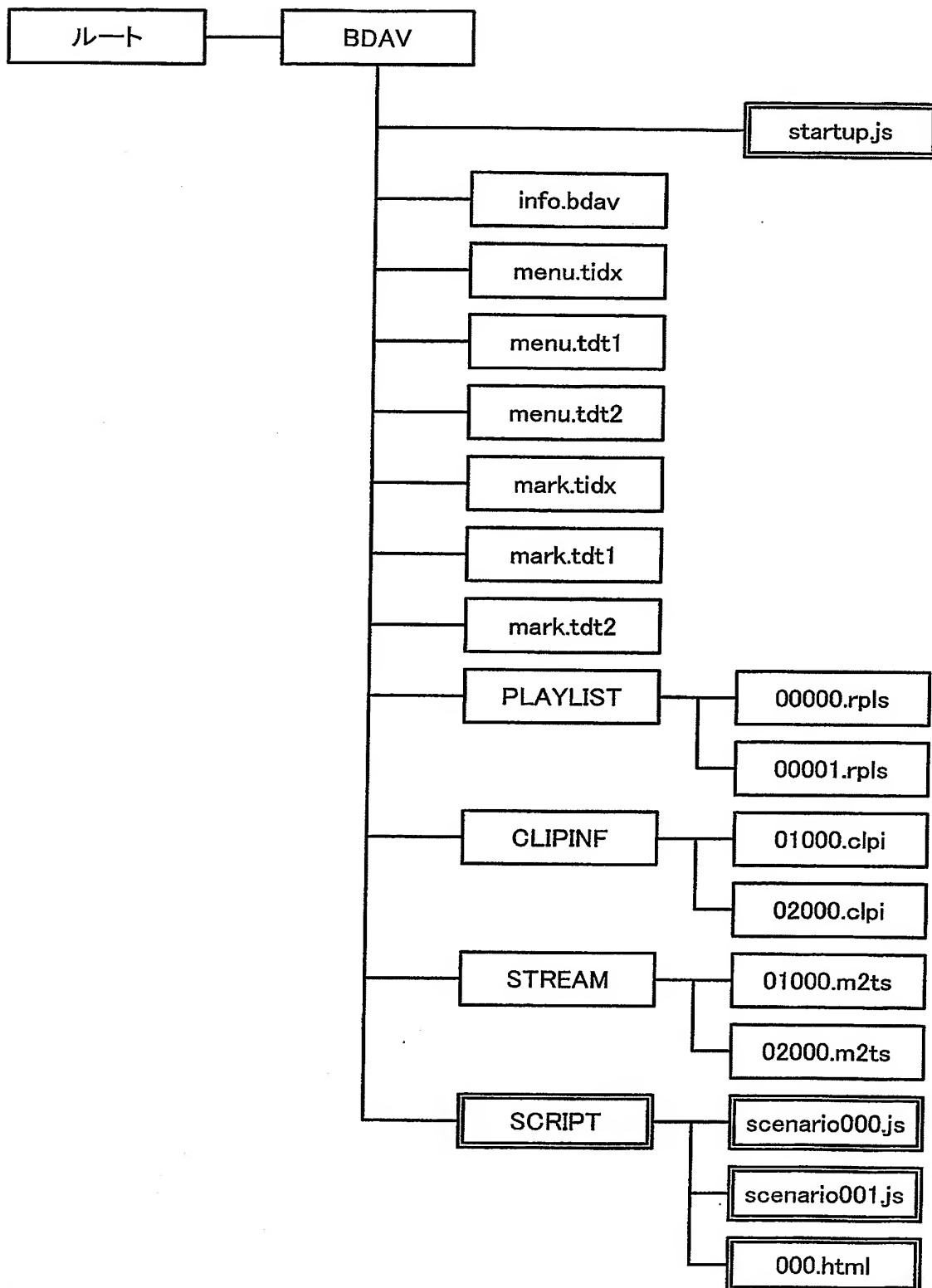
## 第 3 8 図



## 第39図

ファイル名	内容
startup.js	ディスク挿入時に最初に実行されるスクリプトファイル
scenario000.js	シナリオ一覧画面(シナリオ000)の構成情報を記述したスクリプトファイル
000.html	シナリオ一覧画面のレイアウトを記述したhtmlファイル
00000.rpls	シナリオ一覧画面表示中に背景として再生されるプレリストファイル
scenario001.js	シナリオ001の構成情報を記述したスクリプトファイル
00001.rpls	シナリオ001で再生されるプレリストの情報が記述されたプレリストファイル

## 第 4 0 図



## 第 4 1 図

```
=====
startup.js
=====
```

```
function makeArray(n){
    this.length=n;
    for(i=0;i<n;i++){
        this[i]=null;
    }
}
```

```
//シナリオの数と名前の定義
```

```
var scenario=new makeArray(2);
scenario[0]="scenario000";
scenario[1]="scenario001";
```

```
//最初のシナリオ実行
```

```
bdp.playScenario("scenario000");
```

## 第 4 2 図

```
=====
scenario000.js
=====
function UOPControl(){
var keyID=getPressedKey();
  switch(keyID){
    case menu:
      bdp.playScenario("scenario000.js"); //最初に戻る。
      break;
    default:
      break;
  }
}
function playListEnded(){
  bdp.playScenario("scenario000.js"); //最後まで再生し終わったら、また繰り返す。
  return();
}
function MarkEncountered(){ //グラフィクスを読み込むタイミングを示したマーク
  bdp.loadGraphics("000.html","id_1");
  return();
}
function ValidPeriodStarted(f){ //グラフィクスを表示するタイミングを示すマーク
  if(f=="id_1"){
    bdp.showGraphics("id_1");
  }
  return();
}
function ValidPeriodEnded(f){ //グラフィクス表示を停止するタイミングを示すマーク
  if(f=="id_1"){
    bdp.hideGraphics("id_1");
  }
  return();
}
bdp.catchEvent("onKeyPressed","UOPControl()");
bdp.catchEvent("onPlayListEnded","PlayListEnded()");
bdp.catchEvent("onMarkEncountered","MarkEncountered()");
bdp.catchEvent("onValidPeriodStarted","ValidPeriodStarted()");
bdp.catchEvent("onValidPeriodEnded","ValidPeriodEnded()");

bdp.playPlayList("00000.rpls");
```



## 第 4 3 図

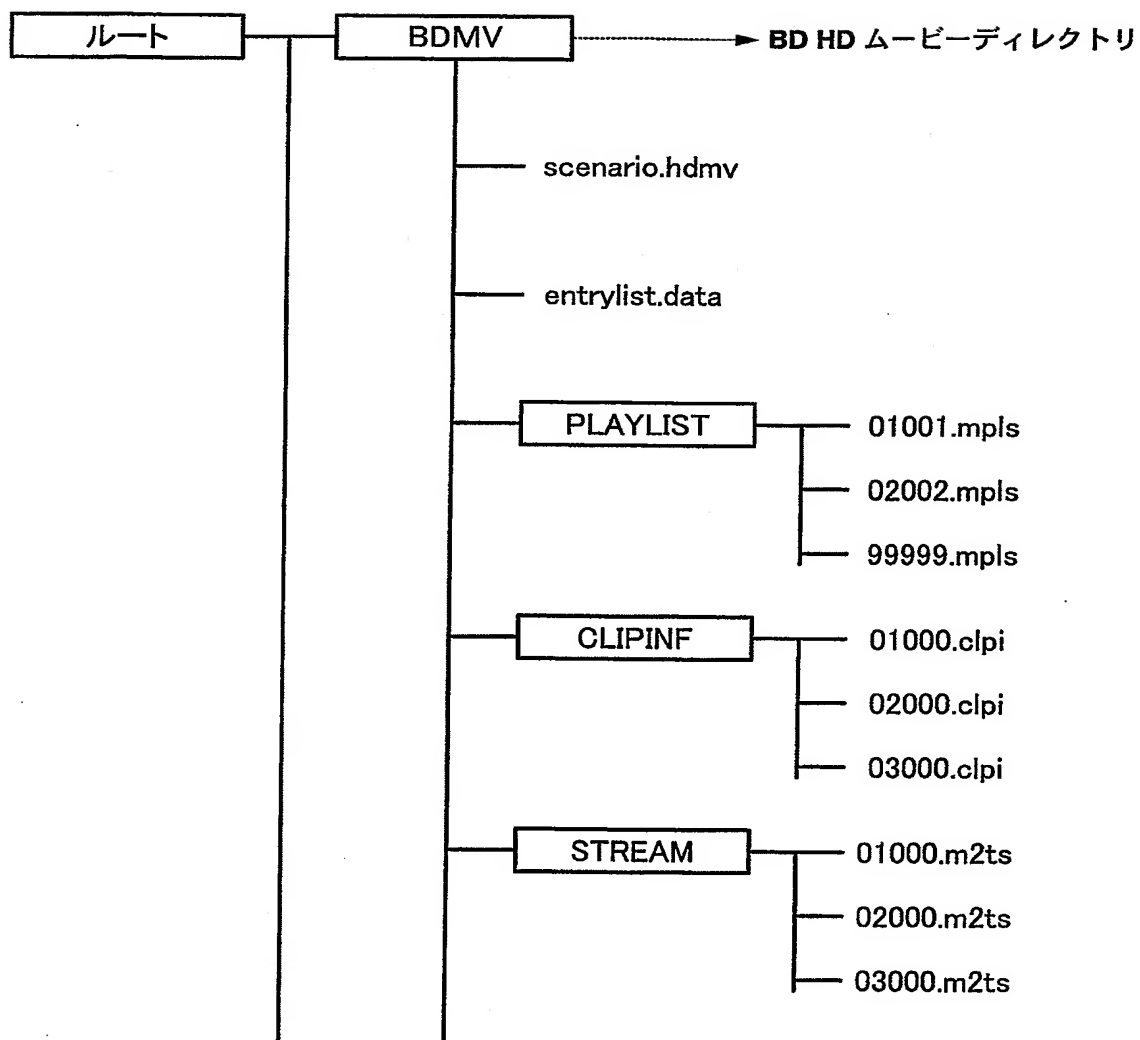
```
=====
000.html
=====
<html>
<head>
<style type="text/css">
<![CDATA[
img#menu {position:absolute;top:200px;left:800px;width:200px;height:50px}
img#scenario001 {position:absolute;top:700px;left:700px;width:400px;height:100px}
]]>
</style>
<script type="text/javascript">
function onMoverhandler(f){
    switch(f){
        case scenario001:
            f.src="201.png";
            break;
        default:
            break;
    }
}
function onMouthandler(f){
    switch(f){
        case scenario001:
            f.src="200.png";
            break;
        default:
            break;
    }
}
function onMclickhandler(f){
    switch(f){
        case scenario001:
            f.src="202.png";
            bdp.playScenario("scenario001.js");
            break;
        default:
            break;
    }
}
</script>
</head>
<body>
    
    
</body>
</html>
```

## 第 4 4 図

```
=====
scenario001.js
=====
function UOPControl(){
    var keyID=getPressedKey();
    switch(keyID){
        case menu: //シナリオ001再生中にメニューキーが押された
            bdp.playScenario("scenario000.js");
            break;
        default;
            break;
    }
}
function playListEnded(){//再生が終了したので、シナリオ000(メニュー画面)に戻る。
    bdp.playScenario("scenario000.js");
    return;
}
bdp.catchEvent("onPlayListEnded","PlayListEnded()");
bdp.catchEvent("onKeyPressed","UOPControl()");

bdp.playPlayList("00001.rpls");
=====
```

## 第45図



## 第 4 6 図

シンタクス	データ長(ビット)	ニーモニック
scenario.hdmv[		
type_indicator	8*4	bslbf
version_number	8*4	bslbf
scenario_start_address	32	
reserved_for_future_use	224	bslbf
Autoplay()		
for(i=0;i<N1;i++){		
padding_word	16	bslbf
}		
Scenario()		
for(i=0;i<N2;i++){		
padding_word	16	bslbf
}		
}		

## 第 4 7 図

シンタクス	データ長(ビット)	ニーモニック
Autoplay(){		
length	32	uimsbf
reserved	16	
number_of_commands	16	
for(i=0;i<number_of_commands;i++){		
command(i)	32	uimsbf
}		
}		

## 第48図

シンタクス	データ長(ビット)	メモリー
Scenario(){		
length	32	
flags	32	
number_of_PlayLists	16	
for(i=0;i<Number_of_PlayLists;i++){		
Pre_Command_start_id	32	
Post_Command_start_id	32	
number_of_Pre_Commands	32	
number_of_Post_Commands	32	
reserved	32	
number_of_PlayItems	32	
for(PlayItem_id=0;PlayItem_id<number_of_PlayItems;PlayItem_id++){		
PI_Command_start_id	32	
number_of_PI_Commands	32	
}		
reserved		
// Command table for each PlayList		
number_of_PL_Commands	16	
for(j=0;j<number_of_PL_Commands;j++){		
PL_Command(j)	32	
}		
}		

## 第 4 9 図

シンタクス	データ長(ビット)	ニーモニック
entrylist.data[		
type_indicator	8*4	bslbf
version_number	8*4	bslbf
ScenarioEntry_start_address	32	uimsbf
reserved_for_future_use	224	bslbf
AppInfo()		
for(i=0;i<N1;i++){		
padding_word	16	bslbf
}		
ScenarioEntry()		
for(i=0;i<N2;i++){		
padding_word	16	bslbf
}		
}		

## 第 5 0 図

シンタクス	データ長(ビット)	ニーモニック
AppInfo()[		
length	32	uimsbf
reserved_for_future_use	16	bslbf
HDMV_name_character_set	8	bslbf
reserved_for_word_align	7	bslbf
PIN_valid_flag	1	bslbf
PIN	8*4	bslbf
// UOP_mask_table() // For directory	64	
HDMV_name_length	8	uimsbf
HDMV_name	8*255	bslbf
]		

## 第51図

シンタックス	データ長(ビット)	ニーモニック
ScenarioEntry{		
length	32	unimbsf
name_character_set	8	bslbf
// Entry PL for the Top Menu		
Top Menu PL{		
flags	32	bslbf
TopMenu_ref_to_PlayList_file_name	8*10	bslbf
TopMenu_ref_to_PlayItem_id	16	unimbsf
TopMenu_name_length	8	unimbsf
TopMenu_name	8*255	bslbf
}		
// Title Entries		
number_of_Titles	16	unimbsf
for(unit_title_number=0;title_number<Number_of_Titles;title_number++){		
flags	32	bslbf
Title_ref_to_PlayList_file_name	8*10	bslbf
Title_ref_to_PlayItem_id	16	unimbsf
Title_name_length	8	unimbsf
Title_name	8*255	bslbf
}		
// Stream Setup Menu for each PL		
number_of_PlayLists	16	unimbsf
for(i=0;i<Number_of_PlayLists;i++){		
SSMenu_flags	32	bslbf
SSMenu_ref_to_PlayList_file_name	8*10	bslbf
SSMenu_ref_to_PlayItem_id	16	unimbsf
}		
}		

## 第 5 2 図

シンタクス	データ長(ビット)	ニーモニック
xxxxx.mpls[		
type_indicator	8*4	bslbf
version_number	8*4	bslbf
PlayList_start_address	32	unimsbf
PlayListMark_start_address	32	unimsbf
reserved_for_future_use	192	bslbf
PLControlInfo()		
for(i=0;i<N1;i++){		
padding_word	16	bslbf
}		
PlayList()		
for(i=0;i<N2;i++){		
padding_word	16	bslbf
}		
PlayListMark()		
for(i=0;i<N3;i++){		
padding_word	16	bslbf
}		
}		



## 第 5 3 図

シンタクス	データ長(ビット)	ニーモニック
PLControlInfo{		
length	32	unimsbf
reserved_for_future_use	8	bslbf
PlayList_character_set	8	unimsbf
reserved_for_future_use	8	
PL_playback_type	8	
if(PL_playback_type==0x2		
PL_playback_type==0x3){		
playback_count	16	
} else {		
reserved_for_word_align	16	
}		
PL_UOP_mask_table() // For PlayList	64	
reserved_for_word_align	8	
PL_random_access_mode	8	
reserved_for_word_align	8	bslbf
PlayList_duration	4*6	bslbf
PlayList_name_length	8	unimsbf
PlayList_name	8*255	bslbf
PlayList_detail_length	16	unimsbf
PlayList_detail	8*1200	bslbf
}		

## 第 5 4 図

PL_playback_type	意味
0x0	リザーブ領域
0x1	プレイアイテムをシーケンシャル再生する(通常再生)
0x2	プレイアイテムをランダム再生する
0x3	プレイアイテムをシャッフル再生する

## 第 5 5 図

PL_random_access_mode	意味
0x0	飛び込み再生・変速再生許可
0x1	飛び込み再生・変速再生禁止

## 第 5 6 図

シンタクス	データ長(ビット)	ニーモニック
PlayList(){		
length	32	unimsbf
number_of_PlayItems	16	unimsbf
number_of_SubPlayItems	16	unimsbf
for(PlayItem_id=0;PlayItem_id<number_of_PlayItems;PlayItem_id++){		
PlayItem()		
}		
for(SubPlayItem_id=0;SubPlayItem_id<number_of_SubPlayItems;SubPlayItem_id++){		
SubPlayItem()		
}		
}		

## 第 5 7 図

シンタクス	データ長(ビット)	ニーモニック
PlayItem(){		
length	16	uimsbf
reserved_for_word_align	8	bslbf
Clip_Information_file_name	8*5	bslbf
Clip_codec_identifier	8*4	bslbf
reserved_for_future_use	7	bslbf
is_multi_angle	1	bslbf
reserved_for_future_use	4	bslbf
connection_condition	4	uimsbf
ref_to_STC_id	8	uimsbf
IN_time	32	uimsbf
OUT_time	32	uimsbf
PI_UOP_mask_table()	64	bslbf
PID_filter()		
reserved_for_word_align	8	bslbf
PI_random_access_mode	8	uimsbf
reserved_for_word_align	8	bslbf
still_mode	8	uimsbf
if(still_mode==0x1){		
still_time	16	uimsbf
} else {		
reserved_for_word_align	16	bslbf
}		
// Angle		
if(is_multi_angle){		
number_of_angles	8	uimsbf
is_seamless_angle_change	8	uimsbf
for(angle_id=1;angle_id<number_of_angles;angle_id++){		
Clip_Information_file_name	8*5	bslbf
ref_to_STC_id	8	uimsbf
IN_time	32	uimsbf
OUT_time	32	uimsbf
}		
}		
}		

## 第 5 8 図

PI_random_access_mode	意味
0x0	飛び込み再生・変速再生許可
0x1	飛び込み再生・変速再生禁止

## 第 5 9 図

still_mode	意味
0x0	Stillなし。
0x1	有限時間のStill。時間は次のstill_timeで指定。
0x2	無限時間のStill。ユーザが解除するまでStillを続ける。
0x3-0xf	予約

## 第 6 0 図

is_seamless_angle_change	意味
0x0	ノンシームレス切り替えのアンゲル
0x1	シームレス切り替え可能なアンゲル

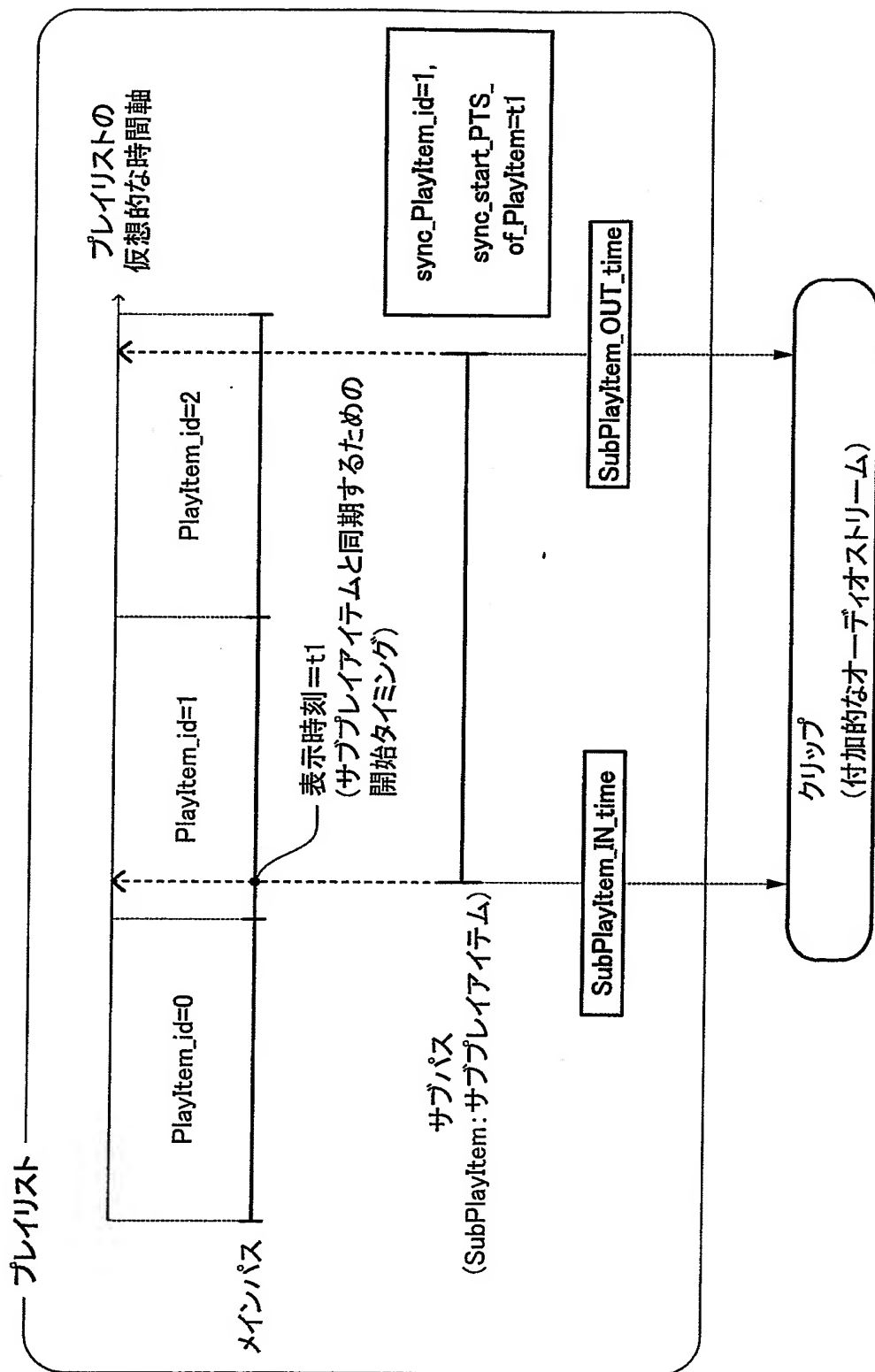
## 第 6 1 図

シンタクス	データ長(ビット)	ニーモニック
SubPlayItem() {		
length	16	unimsbf
Clip_Information_file_name	8*5	bslbf
Clip_codec_identifier	8*4	bslbf
reserved_for_future_use	7	bslbf
is_repeat_flag	1	bslbf
SubPlayItem_type	8	bslbf
ref_to_STC_id	8	unimsbf
SubPlayItem_IN_time	32	unimsbf
SubPlayItem_OUT_time	32	unimsbf
if(is_repeat_flag==0){		
sync_PlayItem_id	16	unimsbf
sync_start_PTS_of_PlayItem	32	unimsbf
} else {		
reserved_for_word_align	16	
reserved_for_word_align	32	
}		
}		

## 第 6 2 図

is_repeat_flag	意味
0	メインパスと同期再生する。
1	メインパスと同期再生しない。再生を繰り返す。

# 第63図





## 第 6 4 図

シンタクス	データ長(ビット)	ニーモニック
zzzzz.clpi {		
type_indicator	8*4	bslbf
version_number	8*4	bslbf
SequenceInfo_start_address	32	uimsbf
ProgramInfo_start_address	32	uimsbf
CPI_start_address	32	uimsbf
ClipMark_start_address	32	uimsbf
reserved_for_future_use	128	bslbf
ClipInfo()		
for(i=0;i<N1;i++){		
padding_word	16	bslbf
}		
SequenceInfo()		
for(i=0;i<N2;i++){		
padding_word	16	bslbf
}		
ProgramInfo()		
for(i=0;i<N3;i++){		
padding_word	16	bslbf
}		
CPI()		
for(i=0;i<N4;i++){		
padding_word	16	bslbf
}		
ClipMark()		
for(i=0;i<N5;i++){		
padding_word	16	bslbf
}		
}		

## 第 6 5 図

シンタクス	データ長(ビット)	ニーモニック
ClipInfo(){		
length	32	unimsbf
reserved	8	bslbf
application_type	8	unimsbf
Clip_stream_type	8	unimsbf
reserved	40	unimsbf
TS_recording_rate	32	unimsbf
num_of_source_packets	32	unimsbf
BD_system_use	1024	bslbf
TS_type_info_block()		
}		

## 第66図

application_type	意味
0	対応するm2tsファイルが、 HDMVトランスポートストリームのルールに従っていない。
1	対応するm2tsファイルが、 HDMVトランスポートストリームのルールに従っている。 (通常のHDMVストリーム)
2	対応するm2tsファイルが、 オーディオ再生に同期する静止画用の HDMVトランスポートストリームのルールに従っている。 (タイムベーススライドショー)
3	対応するm2tsファイルが、 オーディオとは非同期に再生される静止画用の HDMVトランスポートストリームのルールに従っている。 (ブラウザザブルスライドショー)

## 第67図

シンタックス	データ長(ビット)	ニーモニック
SequenceInfo(){		
length	32	unimbsbf
reserved_for_word_align	8	bslbf
num_of_ATC_sequences	8	unimbsbf
for(atc_id=0;atc_id<num_of_ATC_sequences;atc_id++){		
SPN_ATC_start[atc_id]	32	unimbsbf
num_of_STC_sequences[atc_id]	8	unimbsbf
offset_STC_id[atc_id]	8	unimbsbf
for(stc_id=offset_STC_id[atc_id];stc_id<(num_of_STC_sequences[atc_id]+offset_STC_id[atc_id]);stc_id++){		
PCR_PID[atc_id][stc_id]	16	unimbsbf
SPN_STC_start[atc_id][stc_id]	32	unimbsbf
presentation_start_time[atc_id][stc_id]	32	unimbsbf
presentation_end_time[atc_id][stc_id]	32	unimbsbf
}		
}		
}		

## 第68図

シンタックス	データ長(ビット)	ニーモニック
ProgramInfo(){		
length	32	unimsbf
reserved_for_word_align	8	bslbf
num_of_program_sequences	8	unimsbf
for(i=0;i<num_of_program_sequences;i++){		
SPN_program_sequence_start[i]	32	unimsbf
program_map_PID[i]	16	bslbf
num_of_streams_in_ps[i]	8	unimsbf
num_of_groups[i]	8	unimsbf
for(stream_index=0;stream_index<num_of_streams_in_ps[i];stream_index++){		
stream_PID[i][stream_index]	16	unimsbf
StreamCodingInfo(i,stream_index)		
}		
}		

## 第 6 9 図

シンタクス	データ長(ビット)	ニーモニック
StreamCodingInfo(i,stream_index){		
length	8	bslbf
stream_coding_type	8	unimsbf
if(stream_coding_type==0x02){		
video_format	4	unimsbf
frame_rate	4	unimsbf
aspect_ratio	4	unimsbf
reserved_for_word_align	2	bslbf
cc_flag	1	unimsbf
reserved_for_word_align	1	bslbf
}else if(stream_coding_type==0x80//stream_coding_type==0x81//stream_coding_type==0x82){		
audio_presentation_type	4	unimsbf
sampling_frequency	4	unimsbf
language_code	16	bslbf
reserved_for_word_align	8	bslbf
}else if(stream_coding_type==0x90){		
language_code	16	bslbf
T.B.D		
}else if(stream_coding_type==0xA0 ){}		
language_code	16	bslbf
T.B.D		
}		
}		

## 第 7 0 図

シンタクス	データ長(ビット)	ニーモニック
CPI() {		
length	32	unimsbf
reserved_for_word_align	12	bslbf
CPI_type	4	unimsbf
EP_map_for_BDMV()		
}		

## 第 7 1 図

CPI_type	意味
0	将来の使用のために予約
1	EP_mapタイプ
2	TU_mapタイプ
3-7	将来の使用のために予約
8	BDMV用のEP_mapタイプ
9-15	将来の使用のために予約

## 第72図

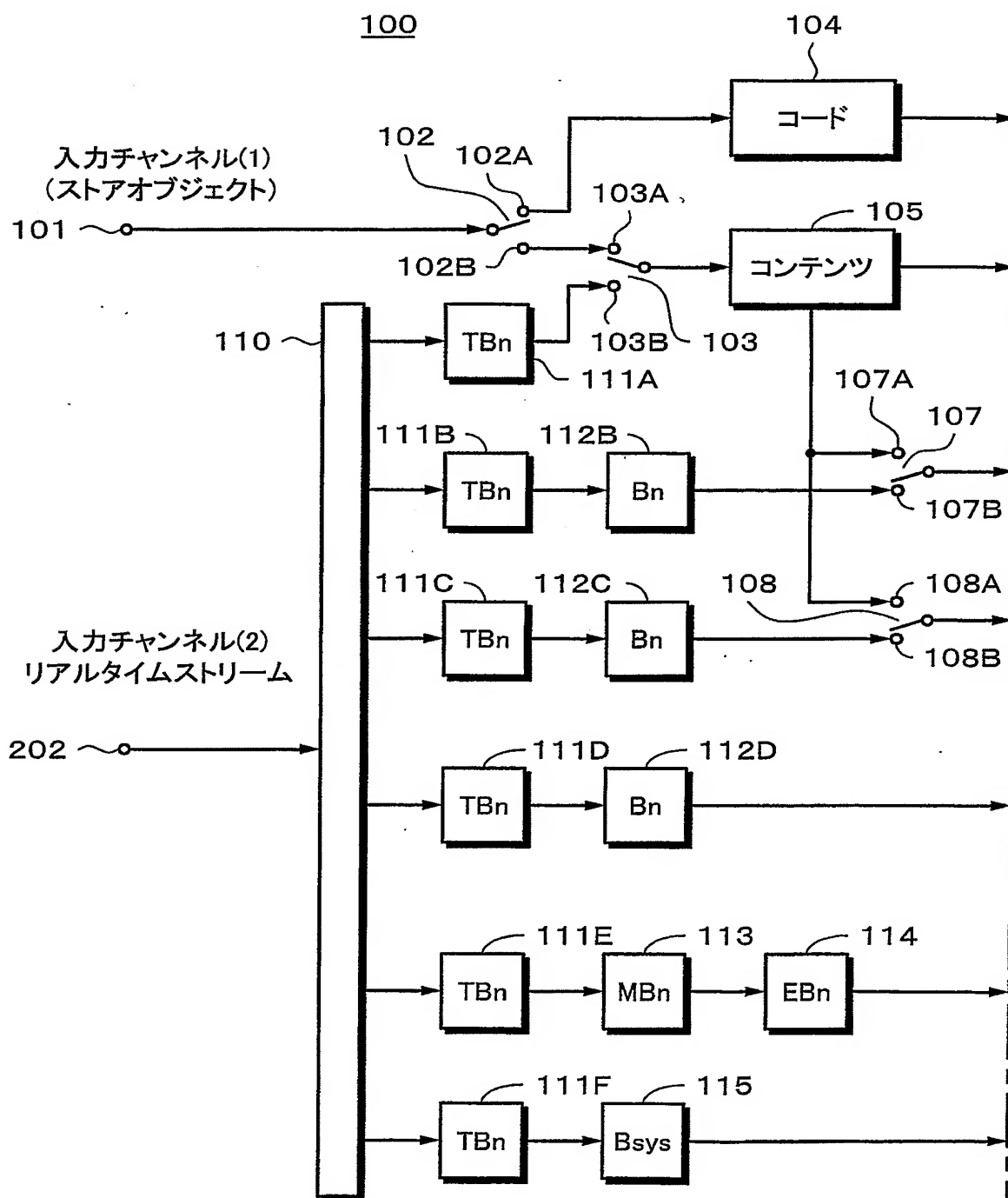
シンタックス	データ長(ビット)	ニーモニック
EP_map_for_one_stream_PID(EP_stream_type, Nc, Nf){		
EP_fine_table_start_address	32	unimbsf
for(i=0; i<Nc; i++){		
ref_to_EP_fine_id[i]	18	unimbsf
PTS_EP_coarse[i]	14	unimbsf
SPN_EP_coarse[i]	32	unimbsf
}		
for(i=0; i<Xc; i++){		
padding_word	16	bslbf
}		
for(EP_fine_id=0; EP_fine_id<Nf; EP_fine_id++){		
is_angle_change_point[EP_fine_id]	1	bslbf
l_end_position_offset[EP_fine_id]	3	bslbf
PTS_EP_fine[EP_fine_id]	11	unimbsf
SPN_EP_fine[EP_fine_id]	17	unimbsf
}		
}		



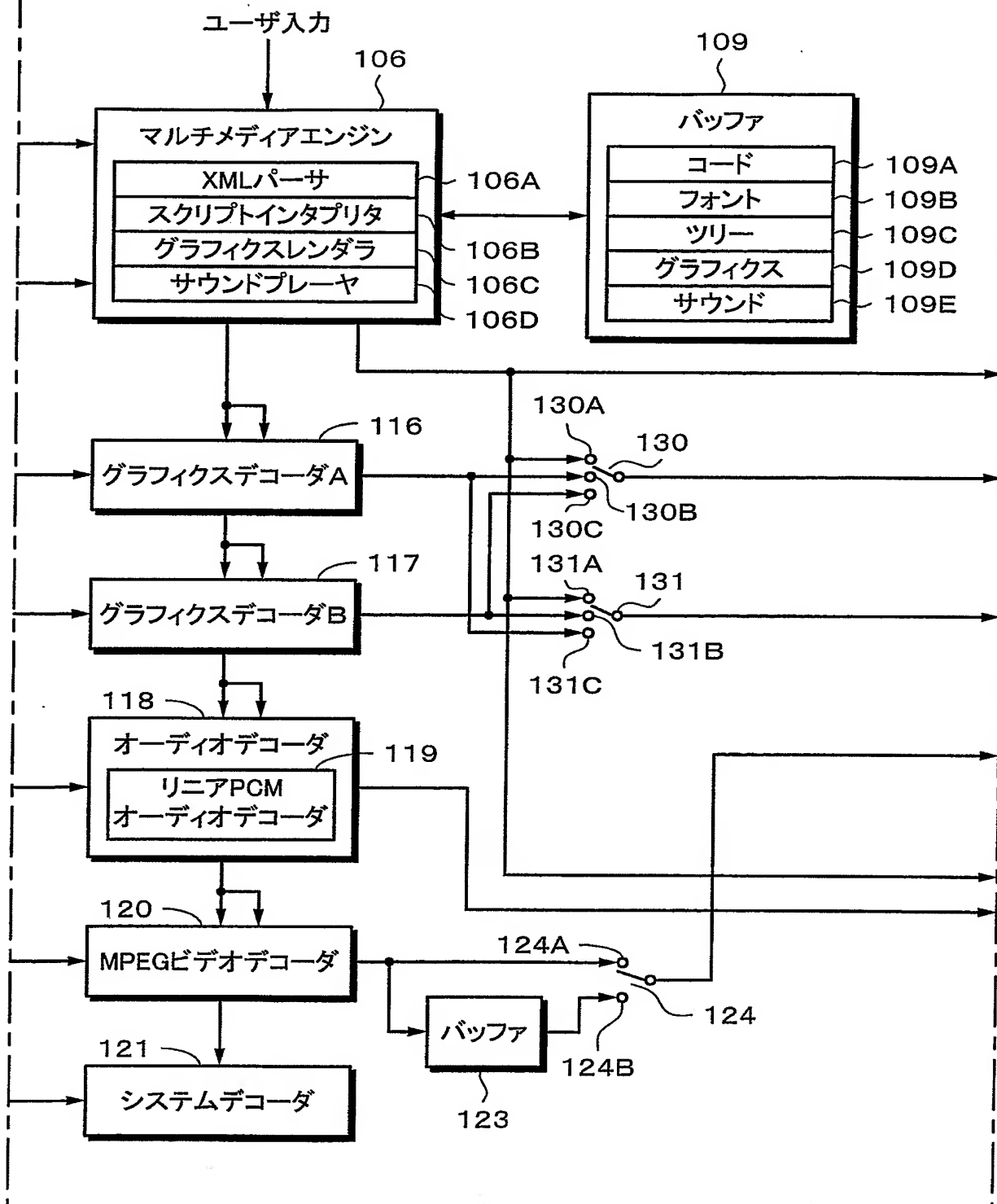
## 第73図A

第73図

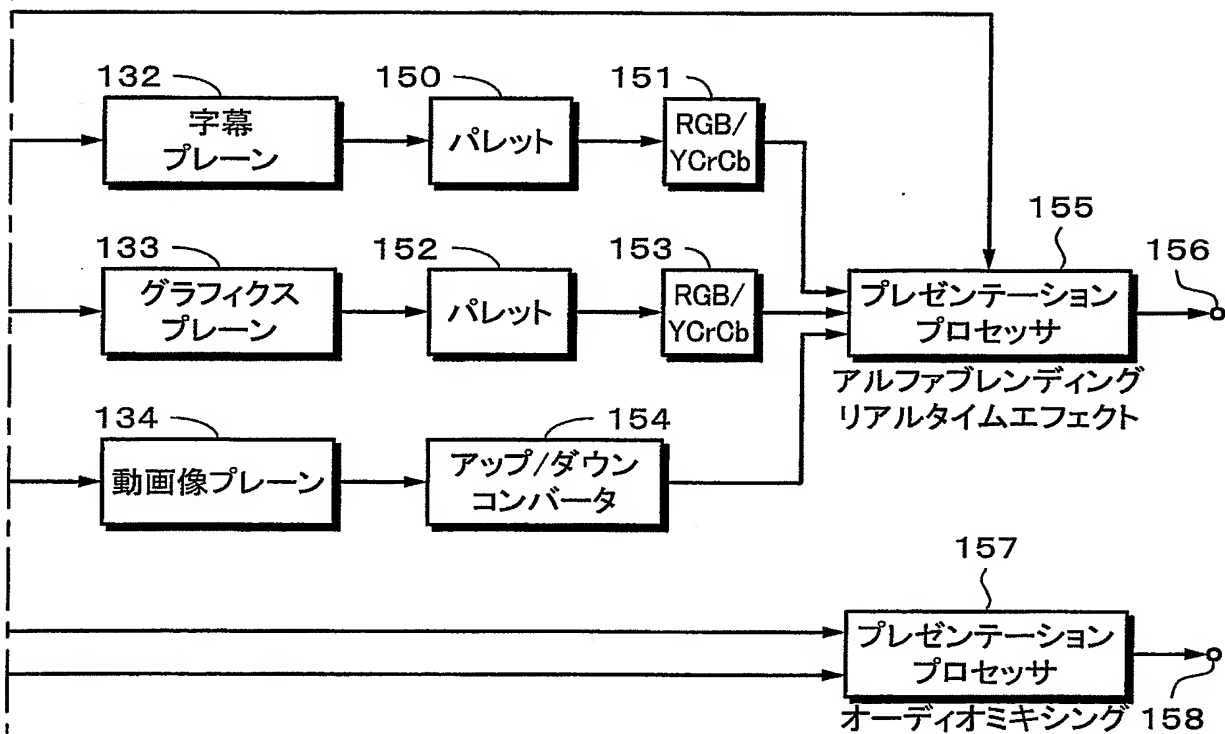
第73図A | 第73図B | 第73図C



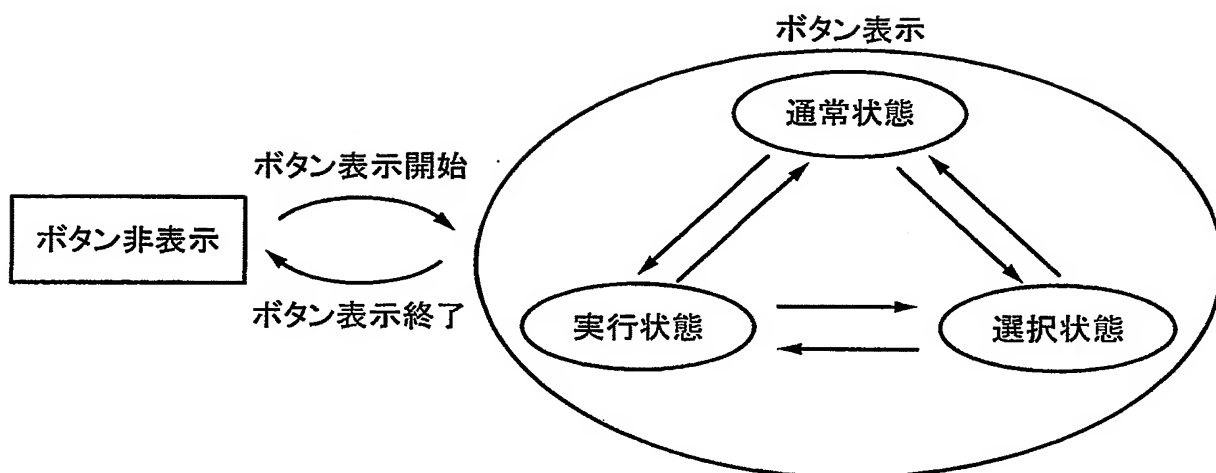
## 第73図B



## 第 7 3 図 C



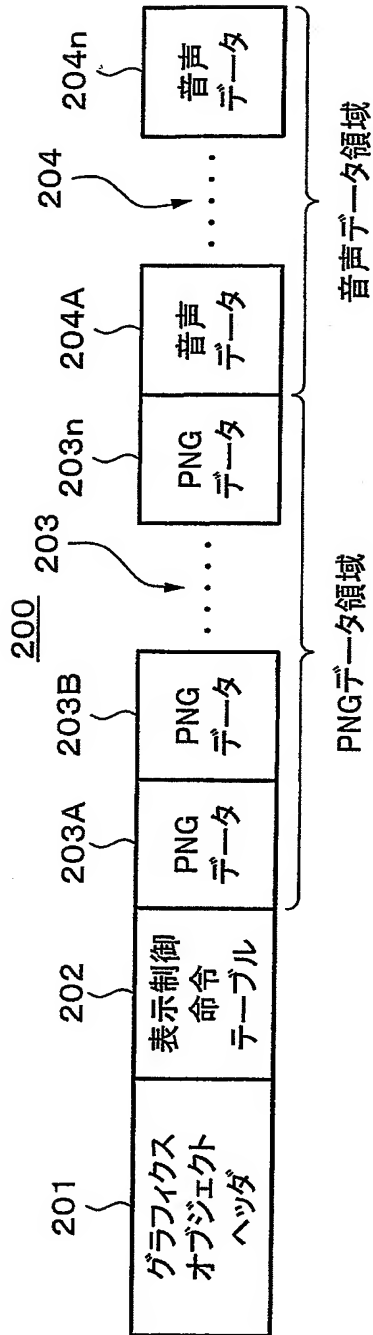
第 7 4 図



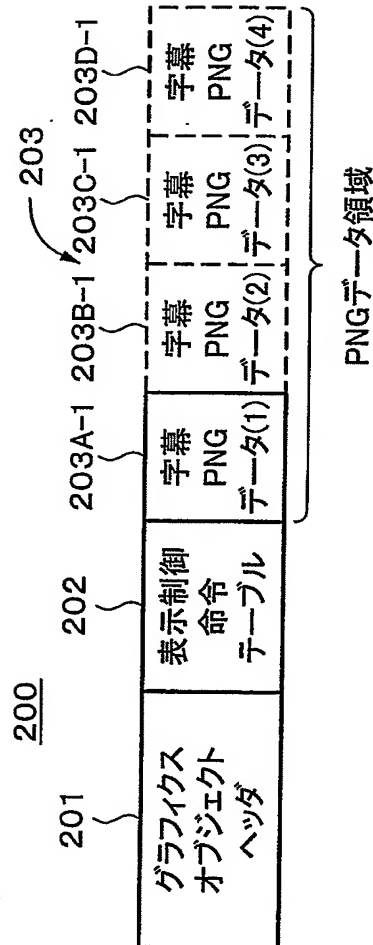
第 7 5 図

オブジェクトの種類	動画プレーンに提示される映像との関係	データ構造	表示先のプレーン
字幕	同期型	グラフィクスオブジェクト (GOBJ)	字幕プレーン
同期型グラフィクス			グラフィクスプレーン
非同期型グラフィクス	非同期型	任意の形式	

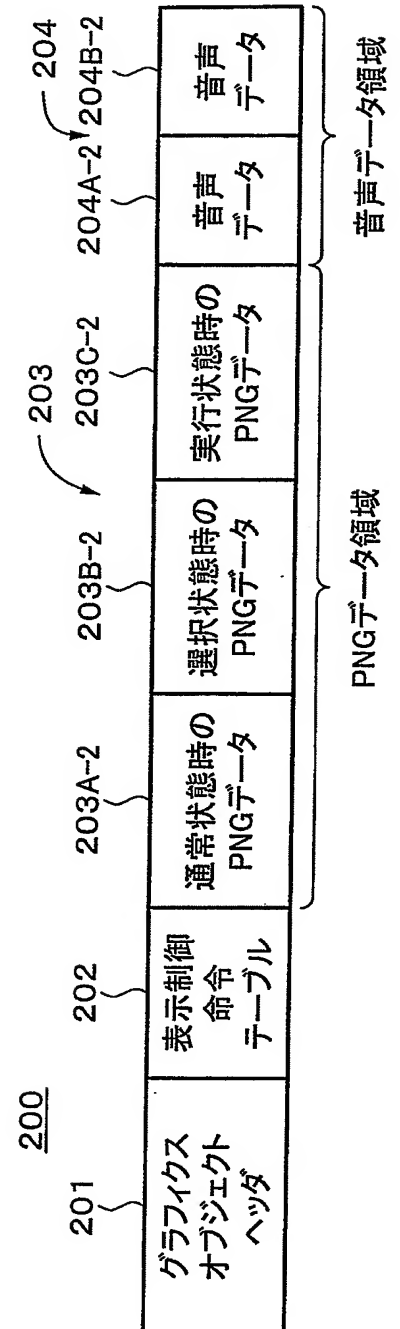
第76図A



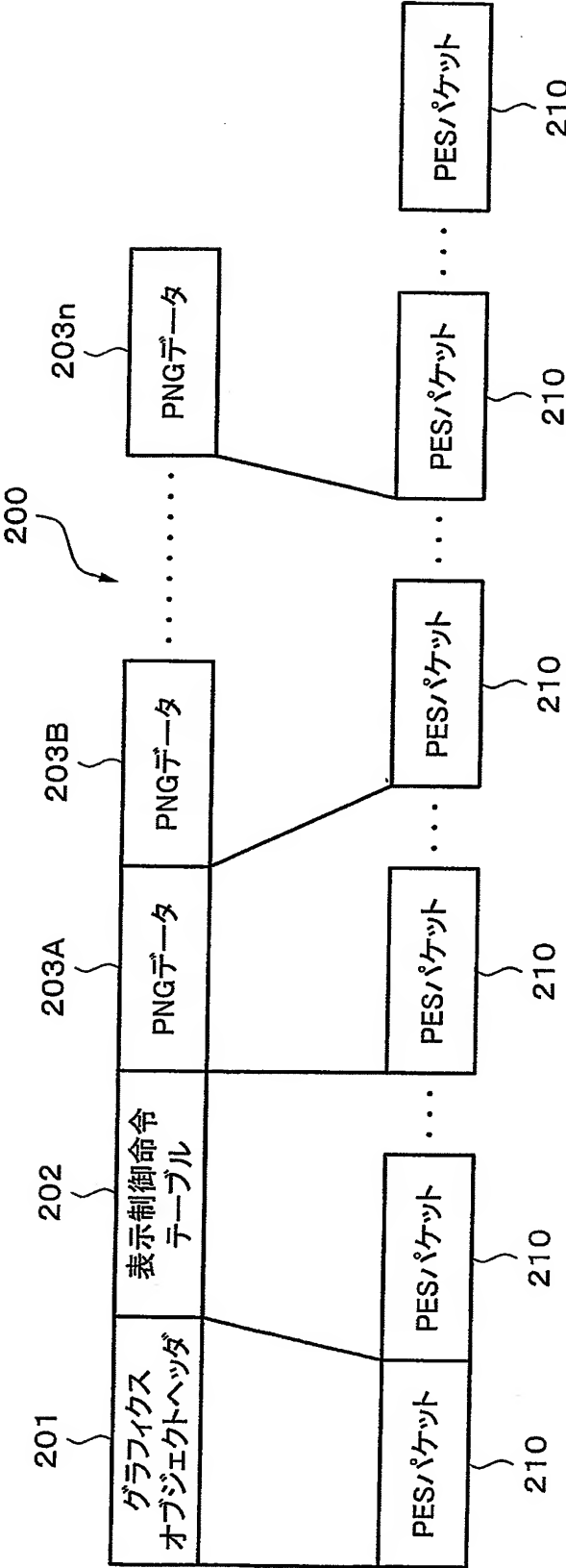
第76図B



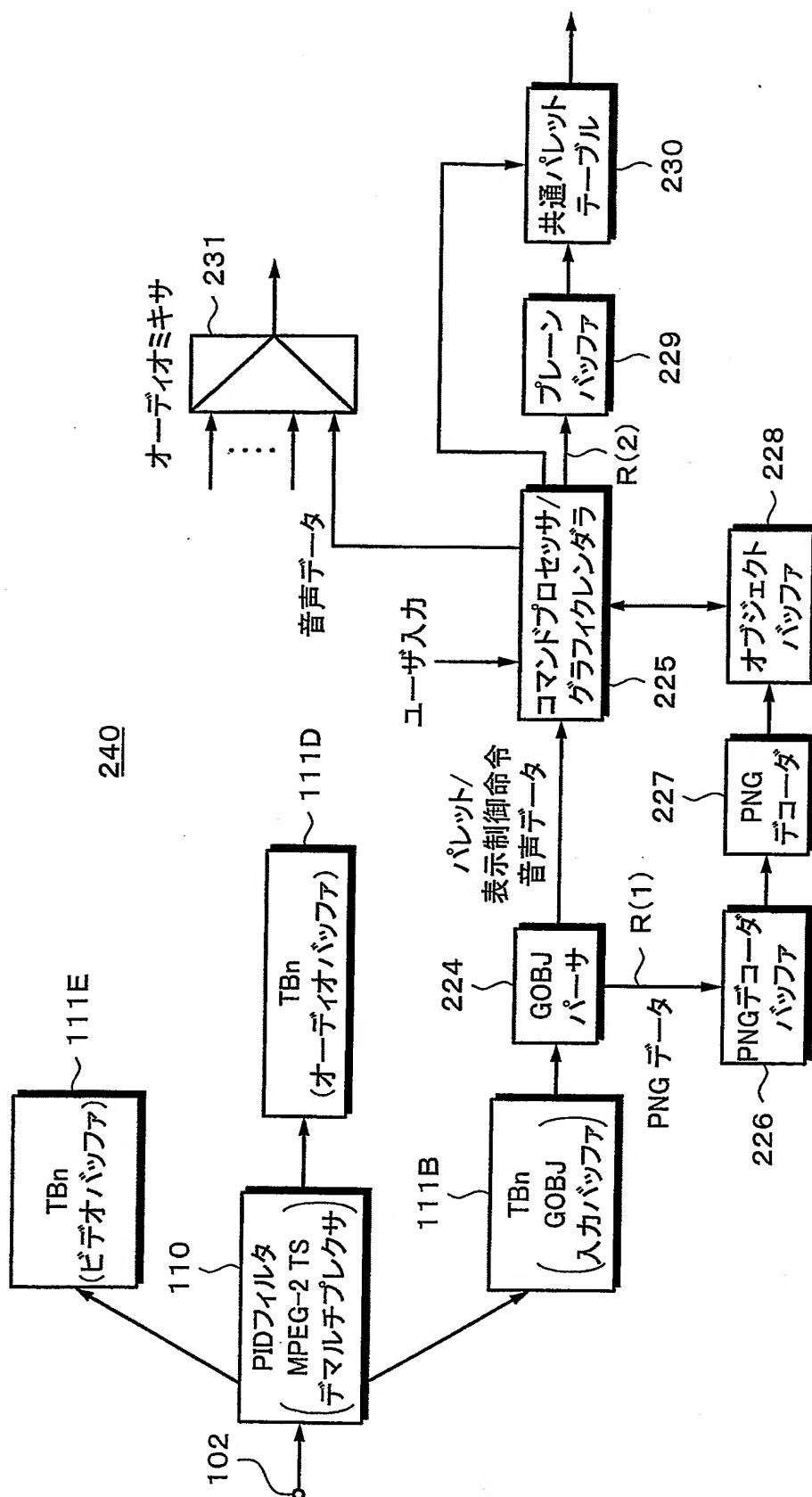
第76図C



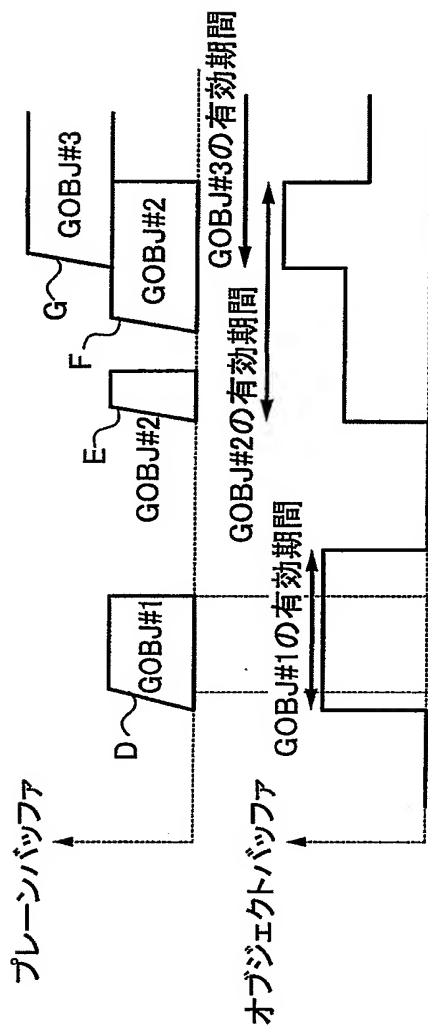
第 7 7 図



第八



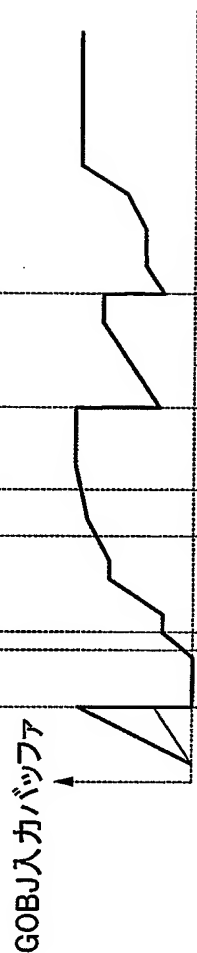
第79図A



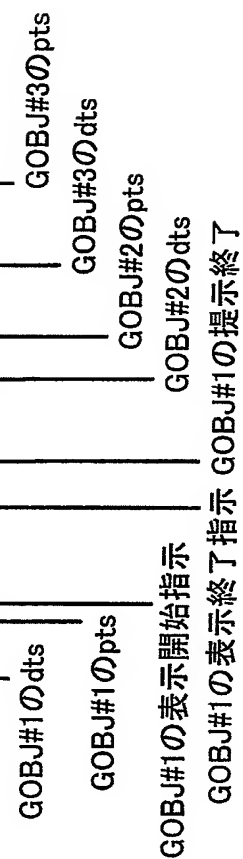
第79図B




第79図C




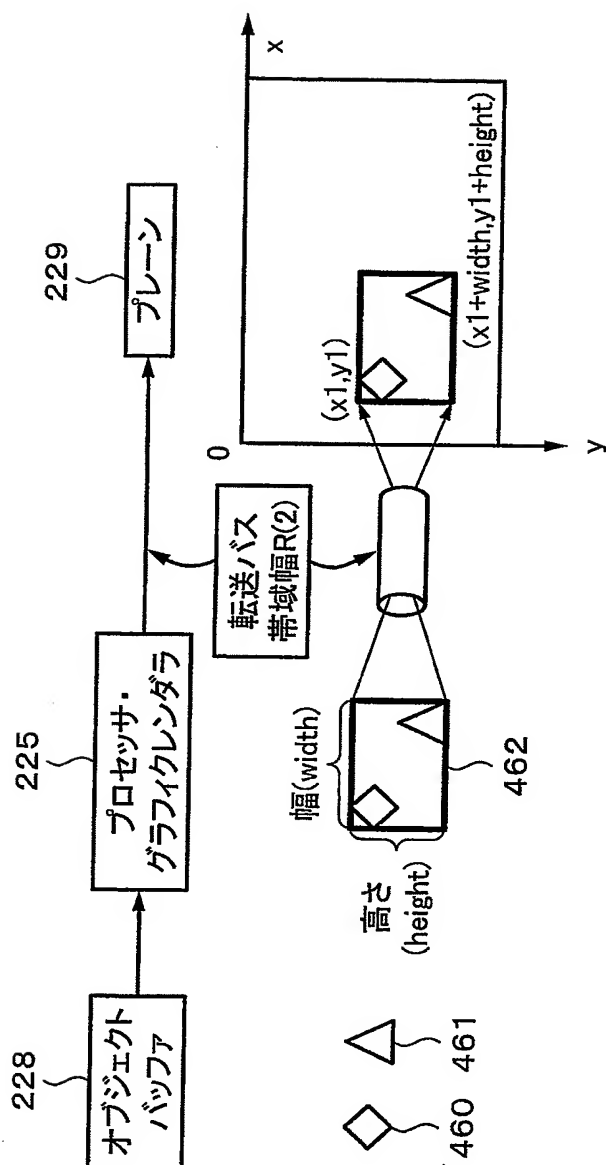
第79図D

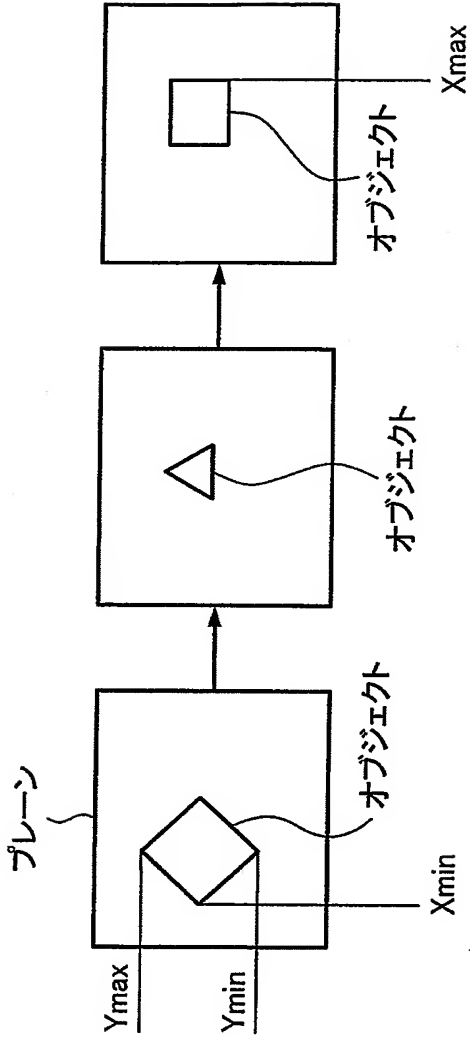




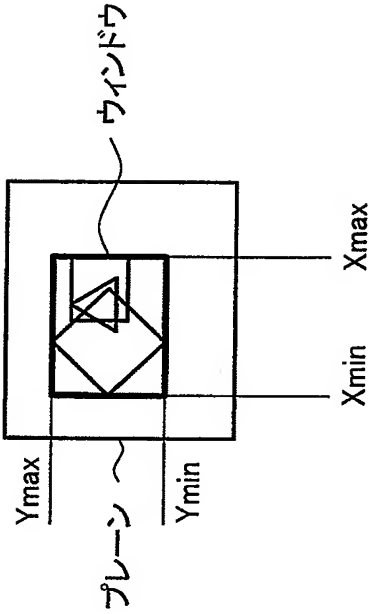


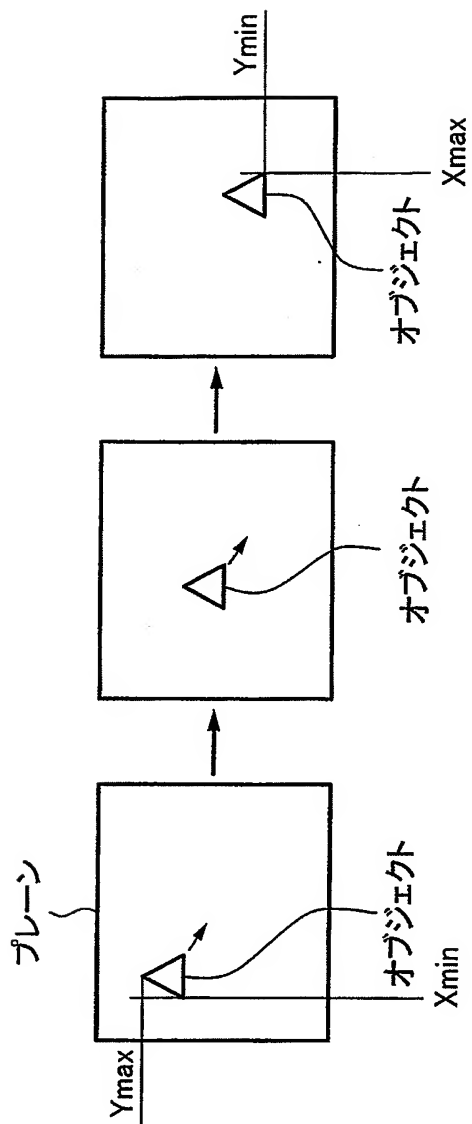




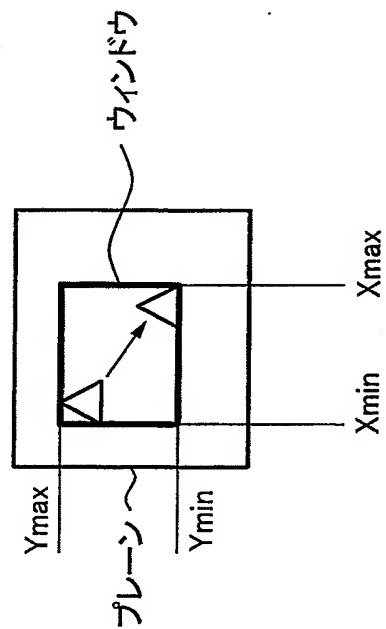
第81図A



第81図B



第82図A



第82図B

## 第 8 3 図

シンタクス	データ長(ビット)	二一モニツク
GraphicsObject(){		
GraphicsObjectHeader(){		
length	8	unimsbf
reserved	7	
presentation_end_time_stamp	33	unimsbf
number_of_PNG_images	8	unimsbf
number_of_Dispcmds	8	unimsbf
GlobalPaletteTable()		
for(i=0;i<Number_of_PNGs;i++){		
start_address_of_PNG_image(i)	32	unimsbf
PNG_file_name(i)	8*32	bslbf
}		
for(=0;i<Number_of_Dispcmds;++){		
start_address_of_PNG_Cmds(i)	32	unimsbf
}		
}		
for(i=0;i<N1;i++){		
padding_word	32	bslbf
}		
GOBJCommandTable(){		
for(i=0;i<Number_of_Dispcmds;i++){		
Dispcmds(i)		
}		
}		
for(i=0;i<N2;i++){		
padding_word	32	bslbf
}		
PNGImageRegion (){		
for(i=0;i<Number_of_PNGs;i++){		
PNG_image(i)		
}		
}		
for(i=0;i<N3;i++){		
padding_word	32	bslbf
}		
SoundDataRegion (){		
}		
for(i=0;i<Number_of_sound_data;i++){		
sound_data(i)		
}		
}		
}		

## 第 8 4 図

シンタクス	データ長(ビット)	ニーモニック
GlobalPaletteTable()[		
reserved_for_future_use	8	bslbf
number_of_palette_entries	8	unimsbf
for(i=0;i<Number_of_palette_entries;i++){		
palette_index_number	8	unimsbf
red_value	8	unimsbf
green_value	8	unimsbf
bule_value	8	unimsbf
alpha	8	unimsbf
}		
}		

## 第85図A

第85図

第85図A

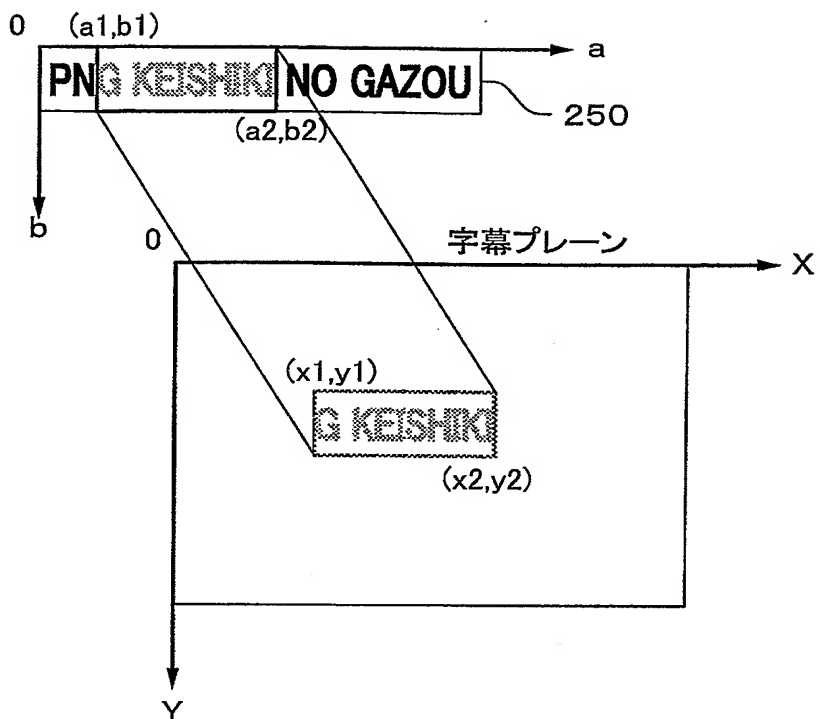
第85図B

表示制御命令	内容
(1)execution_time(start_time)	この命令の後にあり、次のexecution_time(start_time)の前までの命令をstart_timeで指定した時刻に実行することを指示する命令。start_timeの原点は、グラフィクスオブジェクトのpts(presentation time stamp)とする。start_timeの単位はptsと同じ。
(2)fade_in(fade_in_time)	グラフィクスオブジェクトの表示を開始する。現在のパレットテーブルの不透明度の値を徐々に小さくしていくき、fade_in_time後に0(透明)に設定する。fade_in_timeが0の時は、パレットテーブルの色・不透明度で即時に表示される。
(3)fade_out(fade_out_time)	グラフィクスオブジェクトの表示を停止する命令。不透明度の初期値を0から徐々に上げていき、fade_out_time経過後にパレットテーブルの値に設定する。fade_out_timeが0の時は、即時に表示停止する。

## 第 8 5 図 B

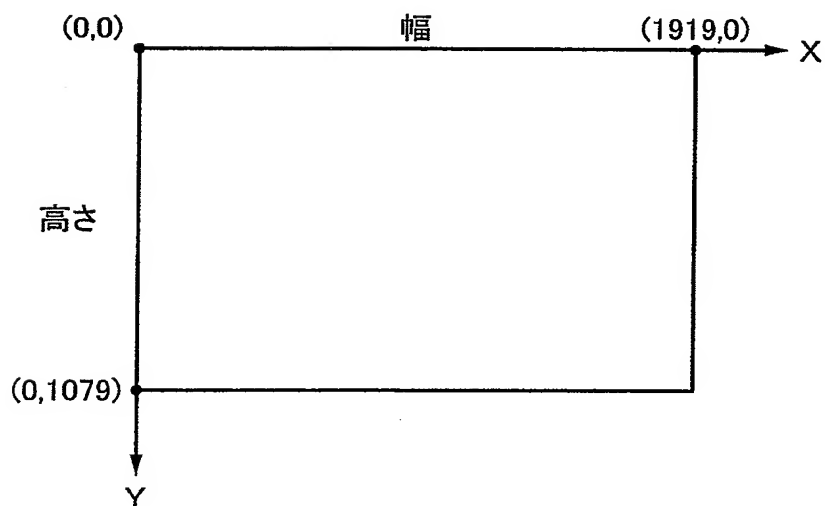
表示制御命令	内容
(4)change_palette (index, newR, newG, newB, newAlpha)	パレット番号indexの色を(newR, newG, newB)に、不透明度をnewAlphaに変更する。
(5)set_display_box(x1,y1,x2,y2)	プレーン上の(x1,y1,x2,y2)で定まる短形の領域にグラフィックスオブジェクトを配置する。
(6)set_clipping_box(a1,b1,a2,b2)	グラフィックスオブジェクト内の(a1,b1,a2,b2)で定まる短形の領域をプレーン上に表示する。
(7)play_sound(sound_id)	sound_idで指定された音声データを再生する。
(8)set_sound(PNG_image_id,sound_id)	PNGデータに音声データを割り当てる命令。PNG_image_idで指定されるPNGデータが表示される際に、sound_idで指定された音声データが再生されるよう指示する命令。PNG_image_idは、シンタックス中のPNG_image(i)におけるi(ループカウンタ)と同じである。

第 8 6 図 A



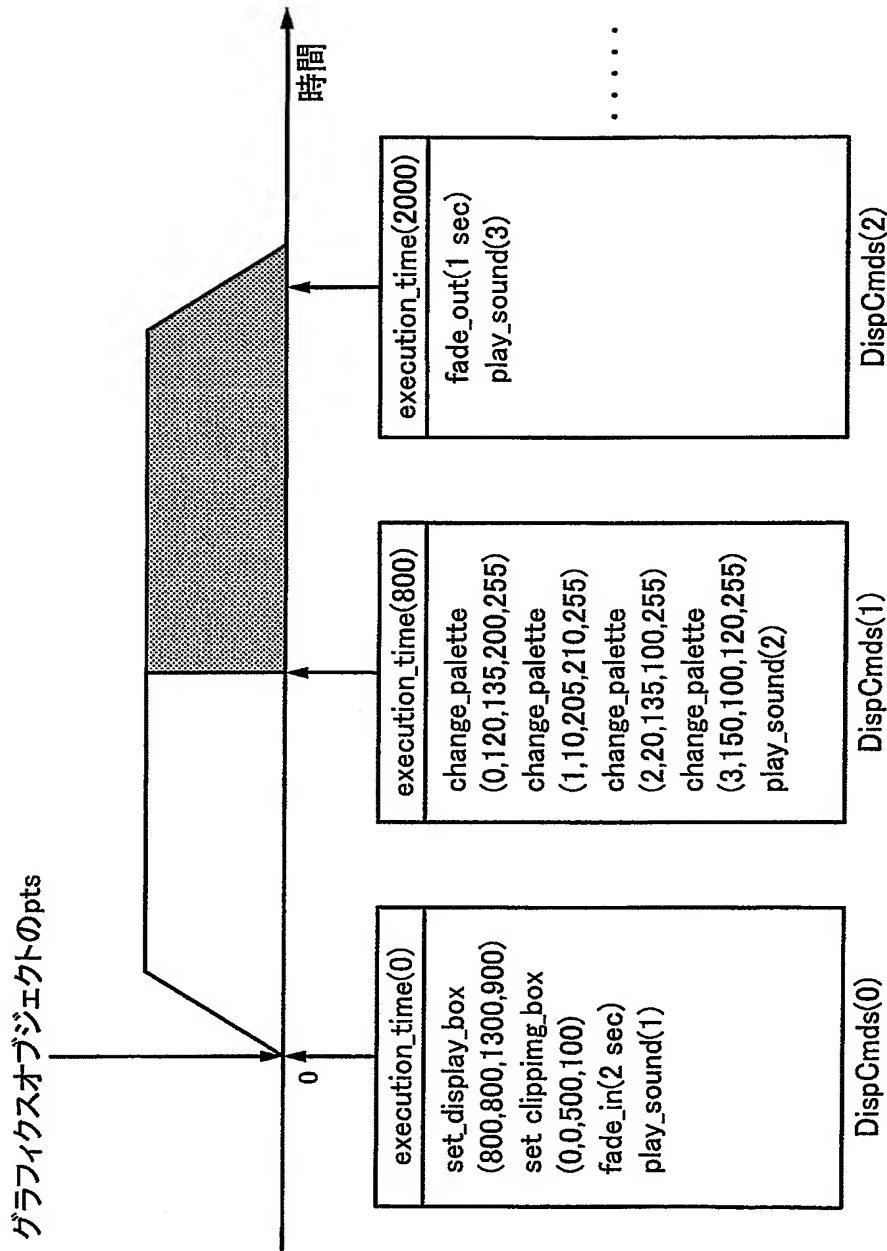
第 8 6 図 B

第 8 7 図





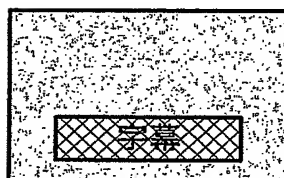
第 8 8 図



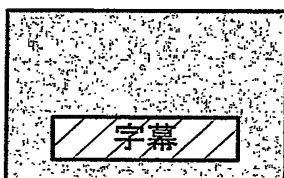
第 8 9 図 A



第 8 9 図 B



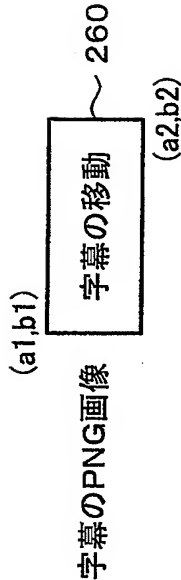
第 8 9 図 C



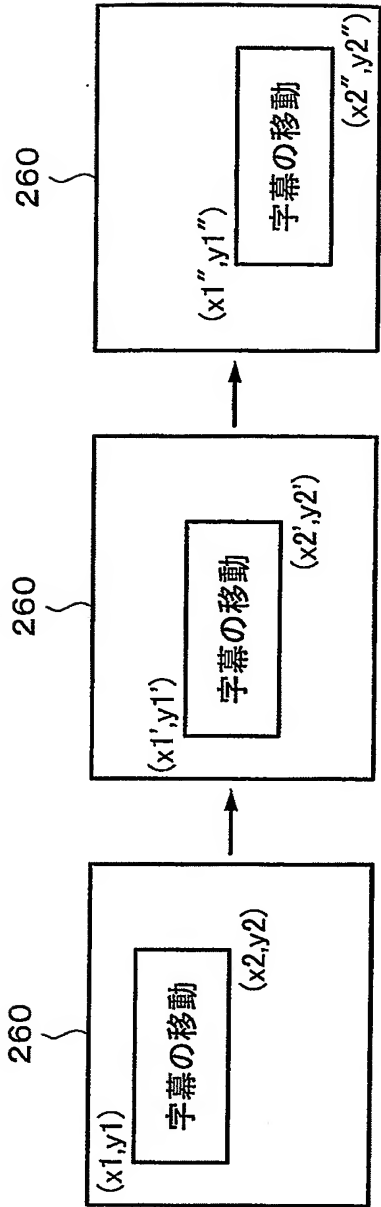
第 8 9 図 D



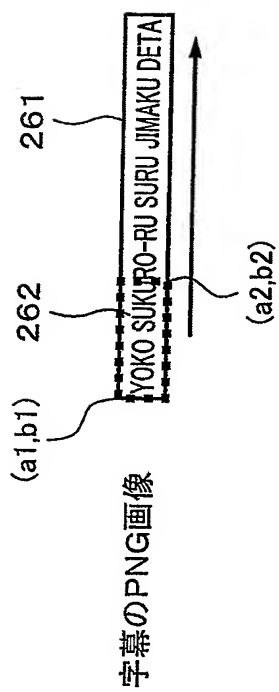
第90図A



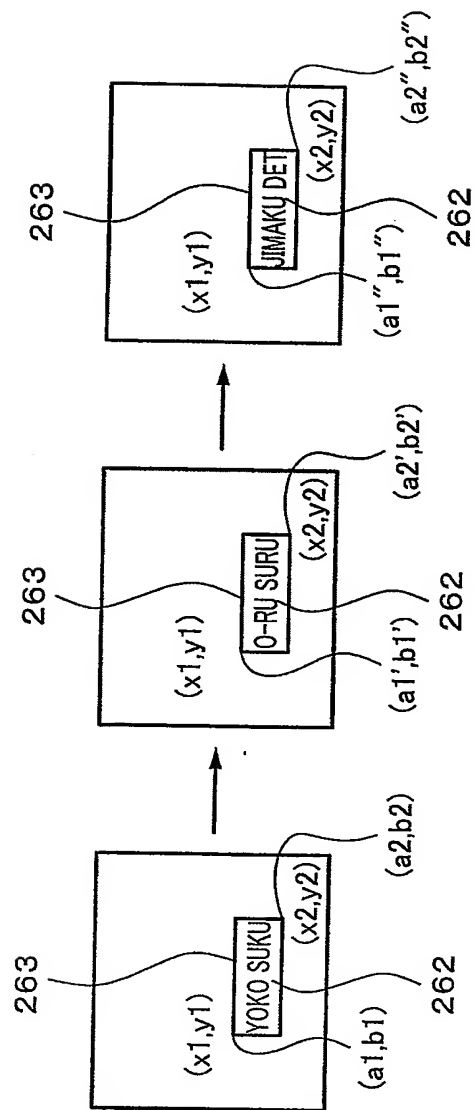
第90図B



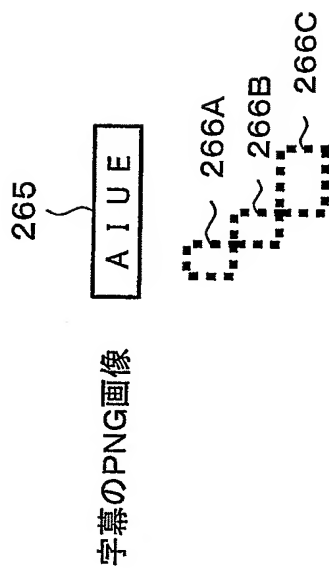
第91図A



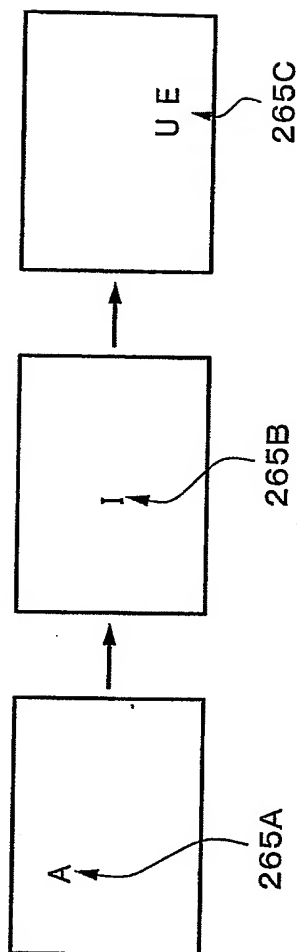
第91図B



第92図A

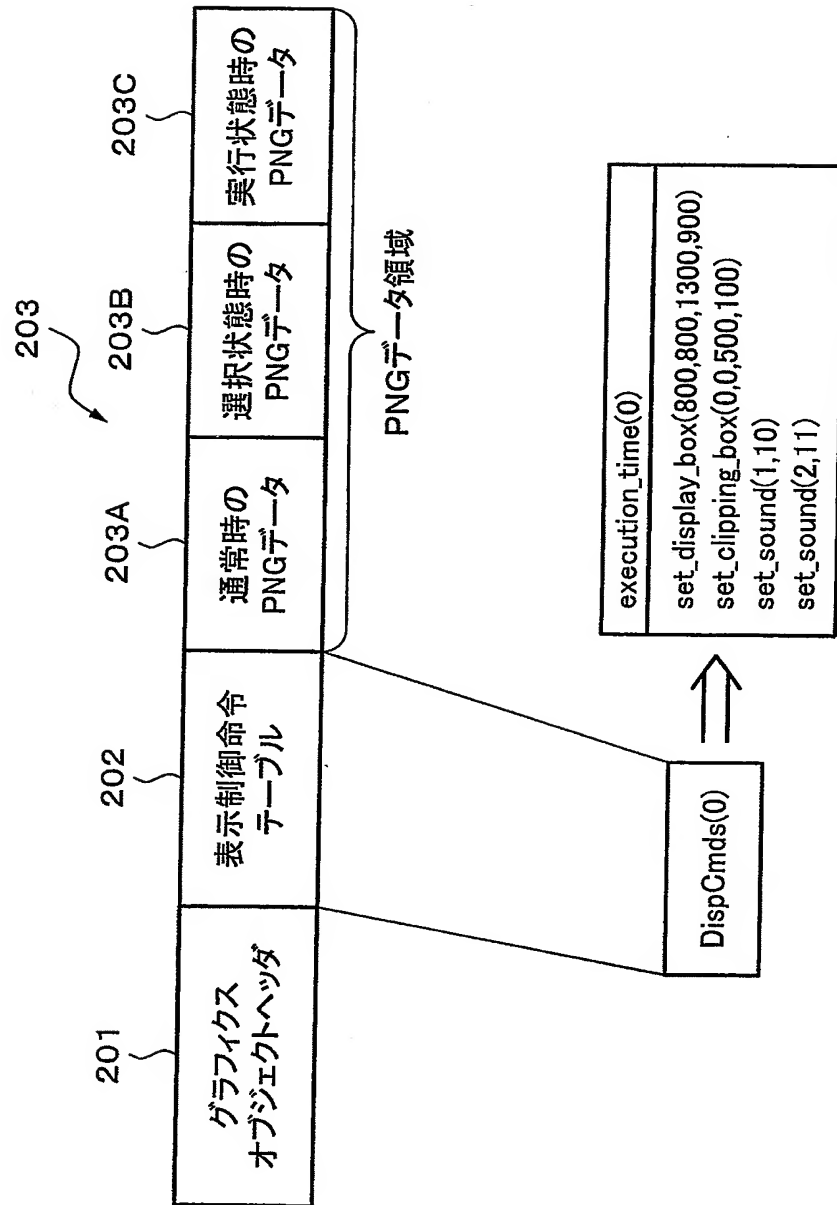


第92図B

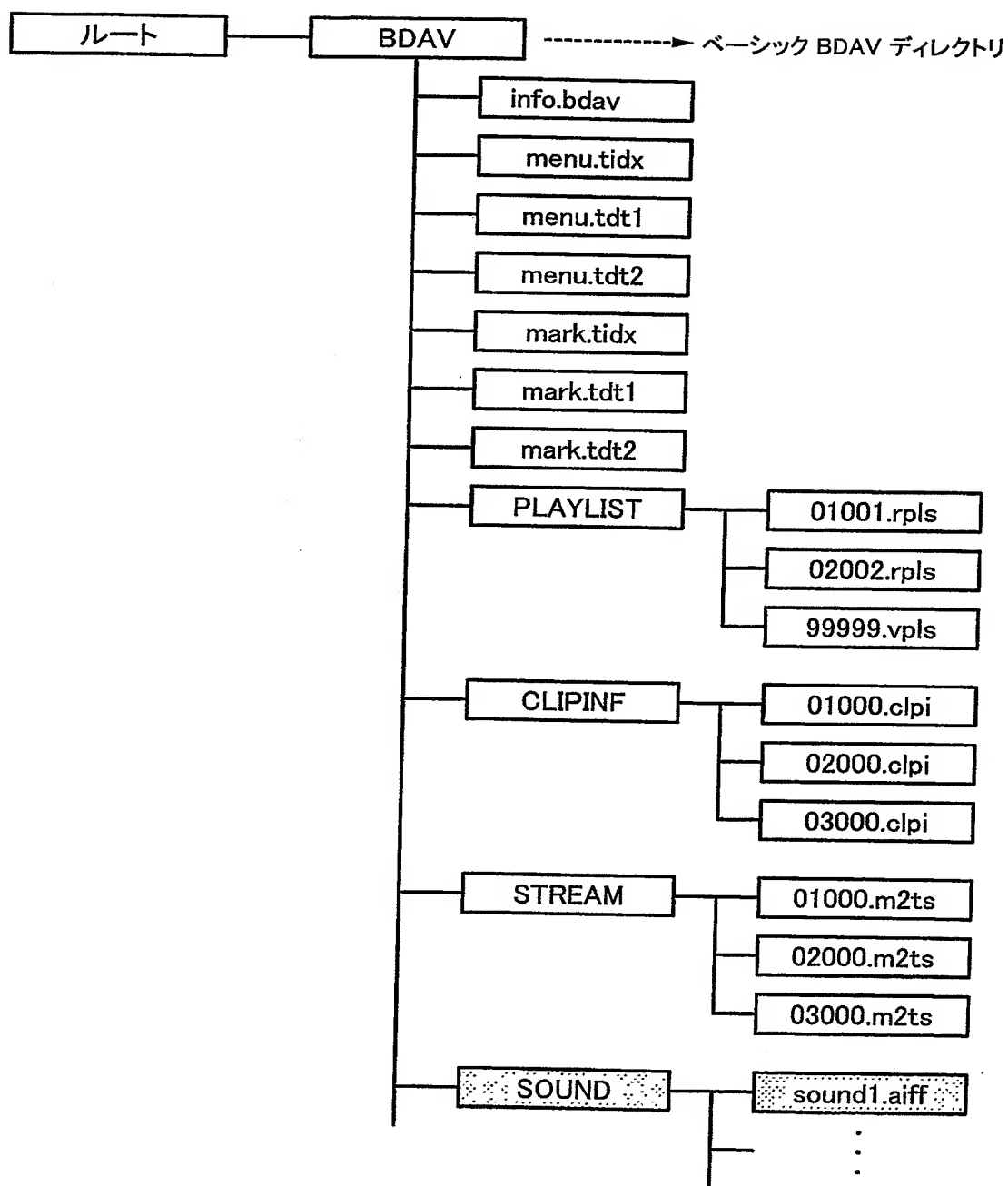


## 第93図

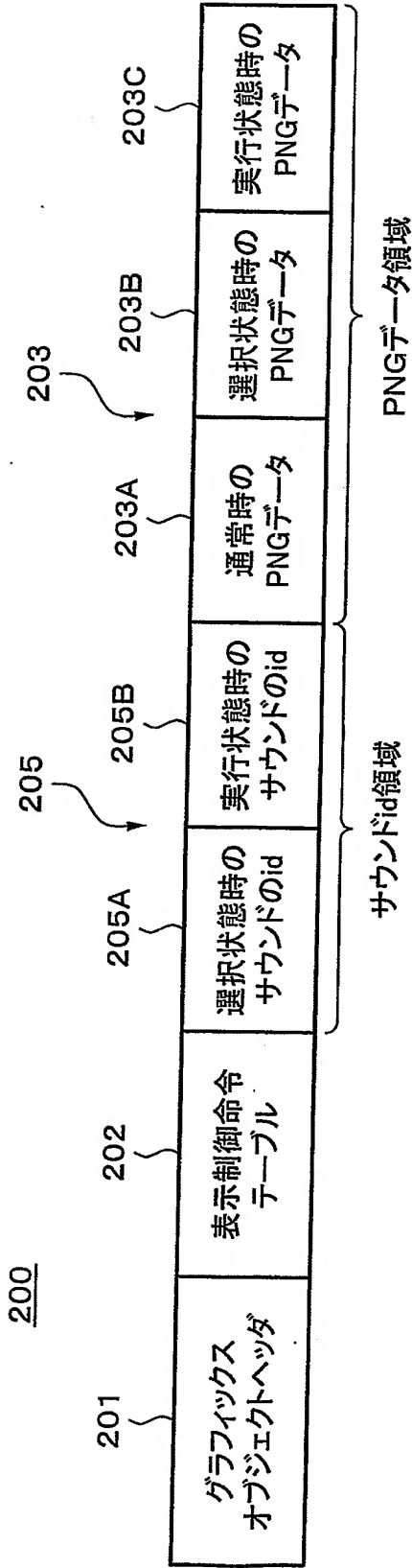
200



## 第 9 4 図

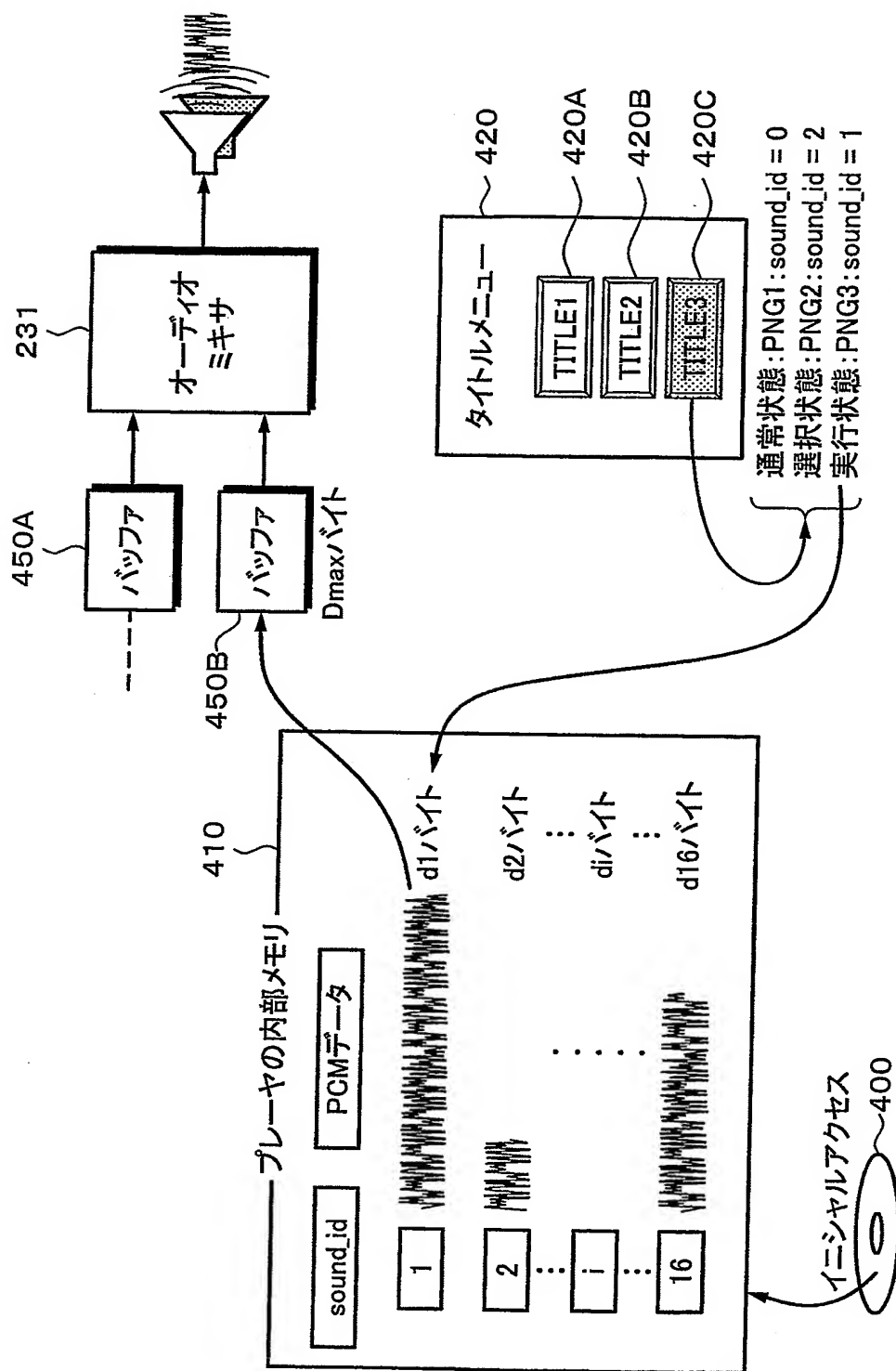


第 9 5 図





## 第96図



## 符 号 の 説 明

- 1 0 動画プレーン
- 1 1 字幕プレーン
- 1 2 グラフィクスプレーン
- 2 2 パレット
- 3 0 B D 仮想プレーヤ
- 3 1 プレーヤコマンド
- 3 2 共通パラメータ
- 4 0 プレイバックコントロールプログラム
- 4 1 メソッド
- 6 0 メニュー画面
- 7 0, 7 0' シナリオ
- 7 3 A ~ 7 3 M プレイリスト
- 1 0 0 プレーヤデコーダ
- 1 0 4 コードバッファ
- 1 0 5 コンテンツバッファ
- 1 0 6 マルチメディアエンジン
- 1 0 9 バッファ
- 1 1 0 P I D フィルタ
- 1 1 6 グラフィックスデコーダ A
- 1 1 7 グラフィックスデコーダ B
- 1 1 8 オーディオデコーダ
- 1 2 0 M P E G ビデオデコーダ
- 1 3 2 字幕プレーン
- 1 3 3 グラフィクスプレーン

- 1 3 4 動画像プレーン
- 2 2 6 PNGデコーダバッファ
- 2 2 7 PNGデコーダ
- 2 2 8 オブジェクトバッファ
- 2 2 9 プレーンバッファ
- 2 3 1 オーディオミキサ

# INTERNATIONAL SEARCH REPORT

International application No.  
PCT/JP03/14511

A. CLASSIFICATION OF SUBJECT MATTER  
Int.Cl<sup>7</sup> H04N5/93

According to International Patent Classification (IPC) or to both national classification and IPC

## B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)  
Int.Cl<sup>7</sup> H04N5/76-5/956, G11B20/10-20/12

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched  
Jitsuyo Shinan Koho 1922-1996 Toroku Jitsuyo Shinan Koho 1994-2004  
Kokai Jitsuyo Shinan Koho 1971-2004 Jitsuyo Shinan Toroku Koho 1996-2004

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

## C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	JP 11-69284 A (Sony Corp.), 09 March, 1999 (09.03.99), Full text; Figs. 1 to 21 & EP 898279 A2 & CN 1212569 A	1-87
A	JP 2001-326910 A (Toshiba Corp.), 22 November, 2001 (22.11.01), Full text; Figs. 1 to 20 (Family: none)	1-87
A	JP 10-308924 A (Sony Corp.), 17 November, 1998 (17.11.98), Full text; Figs. 1 to 9 (Family: none)	1-87

☐ Further documents are listed in the continuation of Box C. ☐ See patent family annex.

* Special categories of cited documents:	"I" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
"A" document defining the general state of the art which is not considered to be of particular relevance	"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
"E" earlier document but published on or after the international filing date	"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)	"&" document member of the same patent family
"O" document referring to an oral disclosure, use, exhibition or other means	
"P" document published prior to the international filing date but later than the priority date claimed	

Date of the actual completion of the international search  
10 February, 2004 (10.02.04)

Date of mailing of the international search report  
24 February, 2004 (24.02.04)

Name and mailing address of the ISA/  
Japanese Patent Office

Authorized officer

Facsimile No.

Telephone No.

## A. 発明の属する分野の分類 (国際特許分類 (IPC))

Int Cl<sup>7</sup> H04N 5/93

## B. 調査を行った分野

調査を行った最小限資料 (国際特許分類 (IPC))

Int Cl<sup>7</sup> H04N 5/76-5/956, G11B 20/10-20/12

最小限資料以外の資料で調査を行った分野に含まれるもの

日本国実用新案公報	1922-1996年
日本国公開実用新案公報	1971-2004年
日本国登録実用新案公報	1994-2004年
日本国実用新案登録公報	1996-2004年

国際調査で使用した電子データベース (データベースの名称、調査に使用した用語)

## C. 関連すると認められる文献

引用文献の カテゴリー*	引用文献名 及び一部の箇所が関連するときは、その関連する箇所の表示	関連する 請求の範囲の番号
A	JP 11-69284 A (ソニー株式会社) 1999.03.09 全文, 第1-21図 & EP 898279 A2 & CN 1212569 A	1-87
A	JP 2001-326910 A (株式会社東芝) 2001.11.22 全文, 第1-20図 (ファミリーなし)	1-87
A	JP 10-308924 A (ソニー株式会社) 1998.11.17 全文, 第1-9図 (ファミリーなし)	1-87

☐ C欄の続きにも文献が列挙されている。☐ パテントファミリーに関する別紙を参照。

## \* 引用文献のカテゴリー

「A」 特に関連のある文献ではなく、一般的技術水準を示すもの  
「E」 国際出願日前の出願または特許であるが、国際出願日以後に公表されたもの  
「L」 優先権主張に疑義を提起する文献又は他の文献の発行日若しくは他の特別な理由を確立するために引用する文献 (理由を付す)  
「O」 口頭による開示、使用、展示等に言及する文献  
「P」 国際出願日前で、かつ優先権の主張の基礎となる出願

の日の後に公表された文献

「T」 国際出願日又は優先日後に公表された文献であって出願と矛盾するものではなく、発明の原理又は理論の理解のために引用するもの  
「X」 特に関連のある文献であって、当該文献のみで発明の新規性又は進歩性がないと考えられるもの  
「Y」 特に関連のある文献であって、当該文献と他の1以上の文献との、当業者にとって自明である組合せによって進歩性がないと考えられるもの  
「&」 同一パテントファミリー文献

国際調査を完了した日

10.02.2004

国際調査報告の発送日

24.2.2004

国際調査機関の名称及びあて先

日本国特許庁 (ISA/JP)  
郵便番号100-8915  
東京都千代田区霞が関三丁目4番3号

特許庁審査官 (権限のある職員)

鈴木 明

5C

9185

電話番号 03-3581-1101 内線 3541